

**Đề cương môn học:
hệ quản trị cơ sở dữ
liệu ngành may**

**CHƯƠNG 1: KHÁI NIỆM CƠ BẢN VỀ CSDL - SỬ DỤNG CÁC CÔNG CỤ
TABLE VÀ QUERY TRONG THAO TÁC CSDL ACCESS**

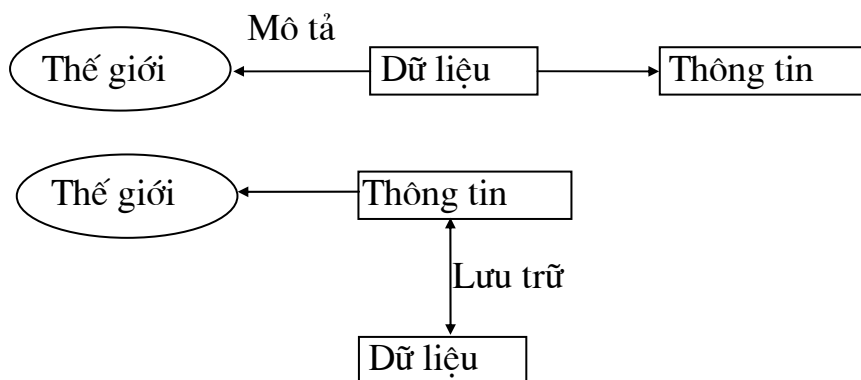
**1.1. CÁC KHÁI NIỆM CƠ BẢN VỀ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG
QUẢN LÝ THÔNG TIN**

1.1.1. Các khái niệm và các thành phần cơ sở dữ liệu

1. Dữ liệu (Data)

- + Dữ liệu là sự mô tả các sự việc, hiện tượng tồn tại trên thế giới.
- + Thông tin (Information): Là sự mô tả, phản ánh về một sự vật, hiện tượng.

Dữ liệu là sự biểu diễn thông tin.



2. File dữ liệu (Data File)

- Trường (Store Field): Là đơn vị nhỏ nhất của dữ liệu, là một trong những giá trị phản ánh bản chất của đối tượng ta quan tâm (còn gọi là thuộc tính).

Ví dụ : Các trường trong File HSSV gồm: MaSV, Hoten, Diachi . . .

- Bản ghi (Store record) : Là tập hợp một số trường xác định đối tượng, hiện tượng.

Ví dụ: Bản ghi Nhân sự bao gồm các trường MaCB, Hoten, Namsinh, Gioitinh, Luong . . . cụ thể như:

001 Nguyen Van A 1970 Nam 350.000 đ

- File dữ liệu (Data File) : File dữ liệu là tập hợp các bản ghi có cùng kiểu loại.

- Biểu diễn dữ liệu : Các dữ liệu được biểu diễn dưới các dạng : số, ký tự, đơn vị đo . . .

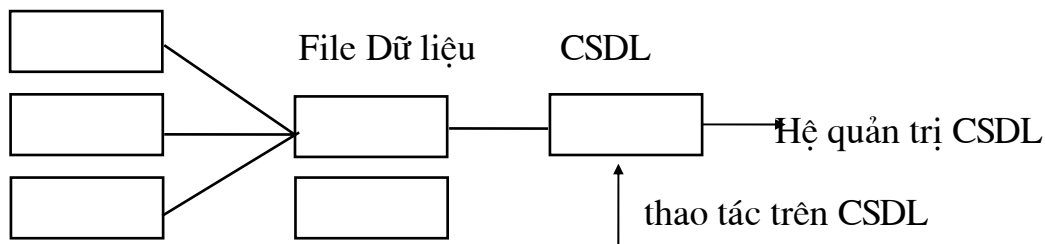
- Mã hoá (Coding) : Là một công cụ biểu diễn dữ liệu theo dạng tóm tắt nhằm mục đích bảo toàn dữ liệu và thao tác dữ liệu nhanh chóng.

Ví dụ : Biển số xe máy 89 A2 1534
Tỉnh Seri Số xe
Hay số điện thoại : 0321 . 910 363
Mã tỉnh mã vùng số máy

3. Cơ sở dữ liệu (DataBase)

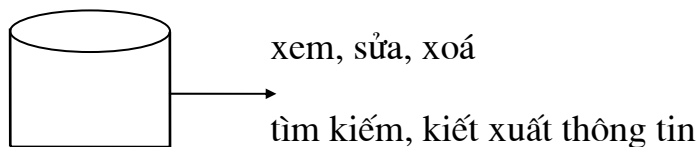
Cơ sở dữ liệu là tập hợp các file được lưu trữ một cách có tổ chức phục vụ cho công việc thao tác và xử lý dữ liệu.

Dữ liệu

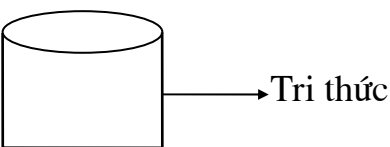


Ví dụ : Hệ quản trị CSDL Foxpro, Access, Visual Basic, Oracle . . .
Xuất phát từ hệ CSDL người ta đã xây dựng được hệ cơ sở tri thức (Knowlegde Base System)

Data Base



Knowlegde



4. Các thành phần của một hệ CSDL :

Bao gồm 4 thành phần sau đây:

- + Dữ liệu (Data): Những thông tin mô tả về sự vật hiện tượng.
- + Phần cứng (Hard Ware): Là các thiết bị mà ở đó lưu trữ được các thông tin.
- + Phần mềm (Soft Ware): Là hệ quản trị CSDL dùng để liên kết giữa mức vật lý và người sử dụng.
- + Người sử dụng (USER): có 3 loại người sử dụng:
 - Người lập trình (Application Programmer) : những người lập các chương trình ứng dụng quản lý CSDL.
 - Người sử dụng cuối (End USER) : Là những người trực tiếp khai thác CSDL.

- Người quản trị CSDL: Bao gồm những người thiết kế, bảo trì CSDL.

1.1.2. Tổng quan về kỹ thuật CSDL

Kỹ thuật CSDL được sử dụng liên quan đến các yếu tố sau:

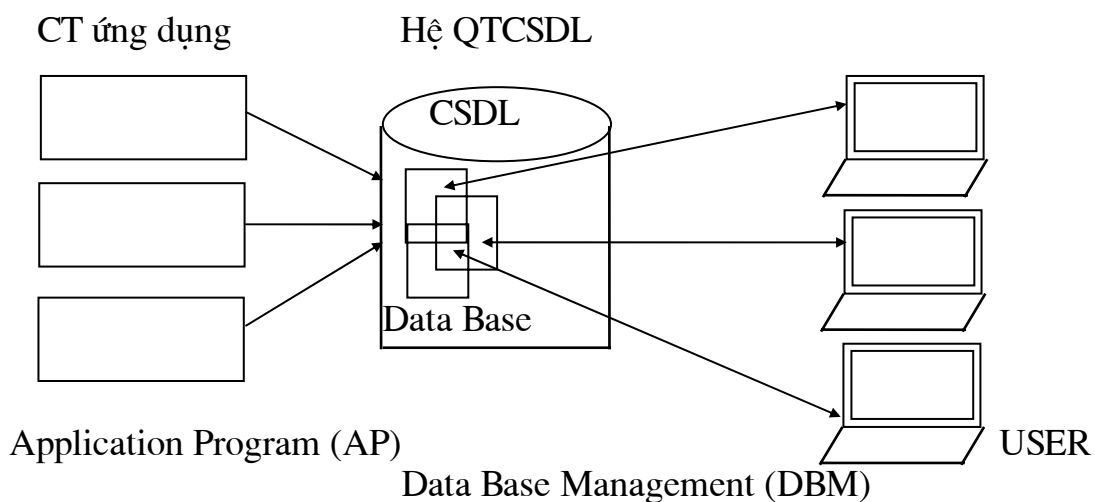
- + Dữ liệu
- + File dữ liệu
- + Mối quan hệ giữa các file dữ liệu
- + Cơ sở dữ liệu
- + Phương thức tổ chức CSDL
- + Các thủ tục lưu trữ, tìm kiếm, xử lý, tổng hợp, đảm bảo an ninh thông tin
- + Kỹ thuật khai thác và bảo trì CSDL

1.1.3. Cấu trúc logic và vật lý

a) Định nghĩa hệ CSDL:

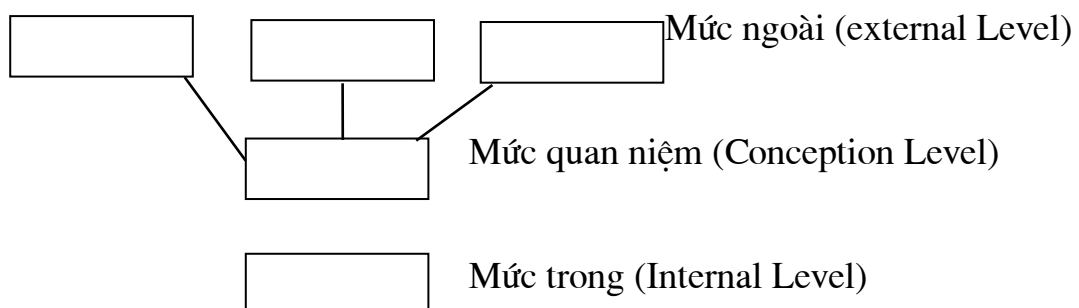
Hệ CSDL là hệ thống lưu trữ các bản ghi ở trong máy tính. Đó là hệ có thể lưu trữ và bảo trì các thông tin lưu trữ.

b) Sơ đồ vật lý của hệ CSDL

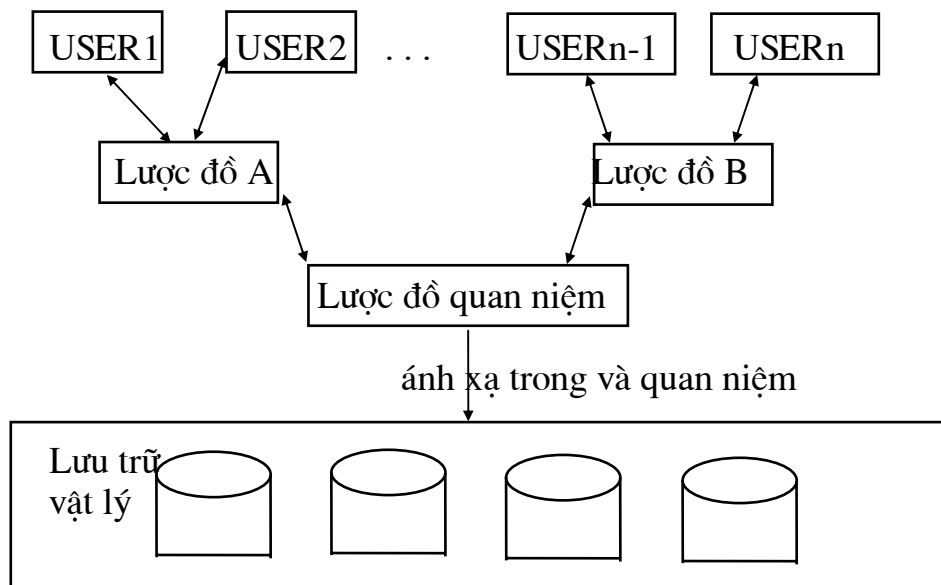


c) Hệ thống CSDL mức logic.

Hệ thống CSDL mức logic là sự biểu diễn CSDL ở mức trừu tượng. Việc biểu diễn này được phân làm 3 mức theo sơ đồ:



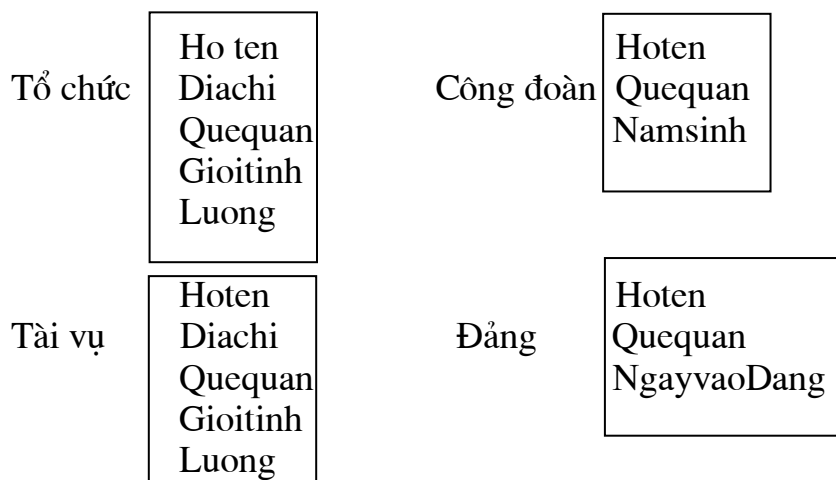
Sơ đồ chi tiết về kiến trúc hệ CSDL: (Data base System Architecture)



d> Một số tính chất đặc trưng của Cơ sở dữ liệu

1- Tính dư thừa (Redundancy).

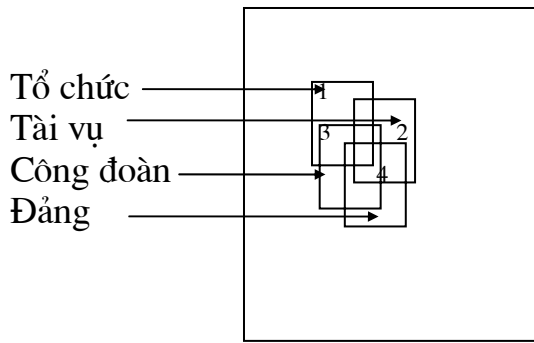
Ví dụ : Xét một hệ thống quản lý nhân sự bao gồm các file:



Từ việc quản lý như vậy dẫn đến một tình trạng rất xấu. Ví dụ ở đây Hoten được lưu trữ ở cả 4 tệp, lương được lưu trữ ở 2-3 nơi. Các tệp Tổ chức, Tài vụ, Công đoàn, Đảng phụ thuộc vào những người sử dụng. Cách thức tổ chức các tệp CSDL như vậy sẽ dẫn đến sự trùng lặp và dư thừa dữ liệu.

- Giải pháp khắc phục:

Tổ chức các dữ liệu thành một kho dữ liệu chung và mỗi người sử dụng, lập trình ứng dụng sẽ sử dụng một phần trong các dữ liệu chung đó tùy theo các ứng dụng của mình.



Giải pháp này được gọi là Share dữ liệu (nghĩa là phân chia các dữ liệu dùng chung).

- 2- *Mâu thuẫn (Inconsistency)*
- 3- *Tính phân chia (Shared)*
- 4- *Tính chuẩn hoá (Standard)*
- 5- *Tính an toàn (Security)*
- 6- *Tính toàn vẹn (Integrity)*
- 7- *Tính đối lập (Conflicting - Requirement)*

1.1.4. Hệ thống quản lý thông tin (MIS – Management Information System)

1. Khái niệm

Hệ thống là tập hợp nhiều thành phần có quan hệ với nhau và thực hiện một nhiệm vụ cụ thể. Một hệ thống gồm nhiều thành phần. Mỗi thành phần thực hiện một chức năng cụ thể của hệ thống.

Ví dụ: Hệ thống trường học bao gồm nhiều trường học cùng thực hiện chung một nhiệm vụ là truyền bá kiến thức tới người học. Nhưng mỗi trường lại có phương thức truyền bá kiến thức tới các đối tượng khác nhau (Học sinh, sinh viên, công nhân...).

Thông thường để biểu diễn hệ thống một cách tổng quan người ta hay sử dụng các mô hình.

Mô hình (Modelling): Là một phương thức biểu diễn hệ thống một cách tổng quát thông qua các hình thức như: dùng lời, sơ đồ...

2. Các hệ thống cơ bản

1- Hệ thống thông tin quản lý:

Là một hệ thống bộ phận của hệ thống kinh doanh thực hiện nhiệm vụ cập nhật, lưu trữ và xử lý thông tin phục vụ cho các hệ thống khác của hệ thống kinh doanh như hệ thống quyết định và hệ thống tác nghiệp.

2- Hệ thống kinh doanh: Là một hệ thống hoạt động trong lĩnh vực kinh doanh với mục đích đem lại lợi nhuận cho toàn bộ hệ thống hay là đem lại lợi nhuận cho doanh nghiệp (nói cách khác là lợi nhuận cho cả những người quản lý và những người thừa hành).

Hệ thống kinh doanh bao gồm các hệ thống bộ phận sau:

+ Hệ thống quyết định:

Là một hệ thống tin tương tác dựa trên một tập hợp nhất các công cụ phần cứng và phần mềm để tạo ra và trình bày thông tin nhằm mục tiêu hỗ trợ cho quản lý trong quá trình làm quyết định. Đó chính là đường lối, sách lược, chiến lược để cho hệ thống kinh doanh hoạt động có hiệu quả nhằm thu được lợi nhuận.

+ Hệ thống tác nghiệp:

Là hệ thống thực hiện nhiệm vụ chính của hệ thống kinh doanh, những thông tin có sẵn cho quyết định ở mức tác nghiệp được xác định rõ ràng trên cơ sở thông tin có sẵn cho người làm quyết định. Hệ thống tác nghiệp là kết quả thực hiện các quyết định sản xuất, đầu tư, kiểm tra.

+ Hệ thống thông tin quản lý:

Đóng vai trò trung gian giữa hệ thống quyết định và hệ thống tác nghiệp với chức năng chủ yếu là thu thập, xử lý dữ liệu và truyền tin.

Hệ thống thông tin bao gồm các thông tin cơ bản của hệ thống như hồ sơ, lý lịch.v.v., nó giúp cho hệ thống quyết định đưa ra phương án tối ưu và giúp cho hệ thống tác nghiệp thực hiện tốt nhiệm vụ của mình.

Vai trò, nhiệm vụ của hệ thống thông tin quản lý:

Đối nội:

Có nhiệm vụ hỗ trợ cho việc thực hiện tác nghiệp tốt hơn, giúp cho hệ thống quyết định đưa ra tư tưởng, phương hướng hoạt động có hiệu quả hơn.

Đối ngoại:

Giúp cho hệ thống khác hoạt động tốt hơn, đồng thời đem lại lợi nhuận cho mình nhiều hơn.

Các thành phần của hệ thống thông tin bao gồm:

+ Con người và phương tiện.

+ Dữ liệu (Data): Bao gồm các thông tin có cấu trúc như: bản ghi...

Các xử lý thông tin: Bao gồm các phương thức xử lý, các chức năng, các quá trình thực hiện các phương pháp.

1.1.5. Các mô hình cơ sở dữ liệu

1. Thực thể - Thuộc tính. Mối quan hệ giữa các thực thể.

a> Thực thể (Entity)

+Thực thể : Tương tự bản ghi

+ Kiểu thực thể: Tương tự File

b> Thuộc tính (Attribute) = trường

Ví dụ : Thực thể Khách mua có các thuộc tính : Taikhoan (MaKH)
Hoten
Diachi
Mahang
.....

* Các loại thuộc tính:

- Thuộc tính khoá:

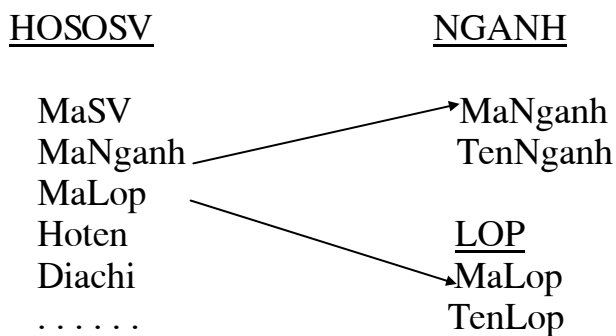
Ví dụ : MaSV trong thực thể HososV là thuộc tính khoá và mỗi sinh viên chỉ có một mã duy nhất.

- Thuộc tính mô tả:

Ví dụ : Thuộc tính Hoten, Diachi trong thực thể HososV là thuộc tính mô tả cho biết họ tên và địa chỉ của một sinh viên cụ thể nào đó.

- Thuộc tính liên kết:

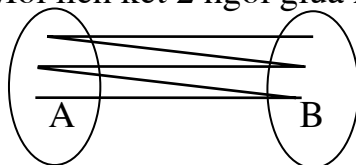
Ví dụ: Xét mối liên hệ giữa 3 kiểu thực thể



c> Mối liên hệ giữa các kiểu thực thể (Entities Relation Ship)

Mối liên kết giữa các thực thể là sự biểu diễn các mối ràng buộc giữa các thực thể hay là sự kết nối giữa 2 hay nhiều thực thể để phản ánh về một sự ràng buộc trong quản lý.

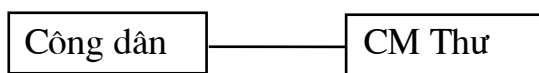
Ví dụ : Mối liên kết 2 ngôi giữa A và B



* Các kiểu liên kết: Bao gồm 3 kiểu liên kết cơ bản :

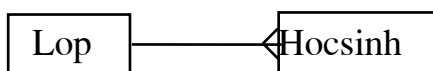
+ Liên kết một - một :

Ví dụ : Mối liên kết giữa thực thể Công dân và thực thể CMthư
Biểu diễn mối liên kết bằng một đường gạch nối 2 thực thể.



+ Liên kết một - nhiều:


Ví dụ : Mối liên hệ giữa 2 thực thể Hocsinh va Lop

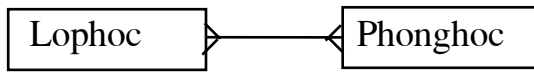


Kí hiệu : —< Một đầu có chân hình tam giác còn đầu kia chỉ có một.

+ Liên kết nhiều - nhiều:

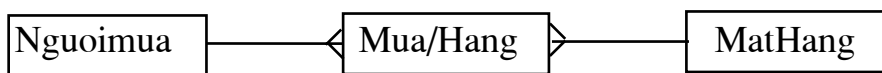
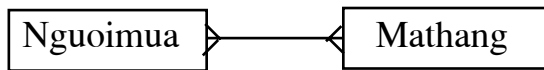
Ví dụ : Mối liên kết giữa Lophoc và Phonghoc. Một Lớp học có thể học nhiều phòng học và ngược lại một Phòng học có thể có nhiều lớp học.

Biểu diễn liên kết:  hai đầu là hai đường gạch nối có chân hình tam giác.



Mỗi liên kết trong liên kết nhiều nhiều thường được thực thể hoá thành các liên kết một nhiều giữa các thực thể.

Ví dụ :

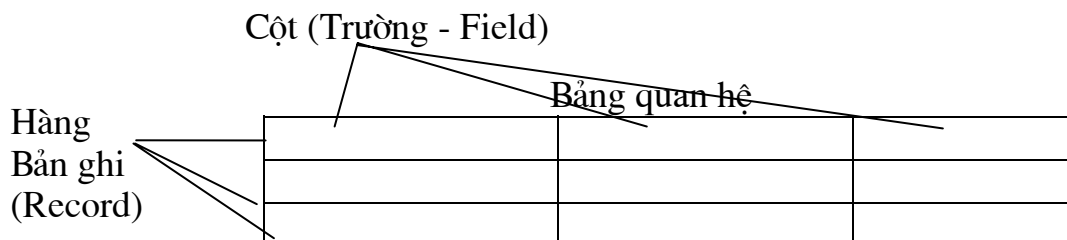


2. Các mô hình dữ liệu (Data Models):

Có 3 loại mô hình dữ liệu:

1- Mô hình quan hệ (Relational Model):

Mô hình quan hệ được xây dựng dựa trên khái niệm tập hợp. Mỗi một quan hệ là một tập hợp gồm có k bộ (tuple), chúng được xây dựng từ các bảng hay nói cách khác là mô hình quan hệ được biểu diễn bằng các bảng có dạng:



Gồm bảng có cột biểu diễn trường (field) và hàng biểu diễn bản ghi (Record).

Tập hợp quan hệ k bộ -> k bản ghi.

Ví dụ: Giả sử ta đang xét một mô hình kinh doanh gồm có:

Chủ hàng - Hàng được thể hiện bởi các bảng sau:

Chủ hàng (S)

S #	S name	Status	City
S1	Smith	20	London
S2	John	10	Pari
S3	Blake	30	Pari

Hàng (P)

Đề cương học phần: **HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU NGÀNH MAY**

P#	Pname	Color	Weight	City
P1	Nut	Red	12	London
P2	Bolt	Green	17	Pari
P3	Crew	Blue	17	Rom
P4	Crew	Red	14	London

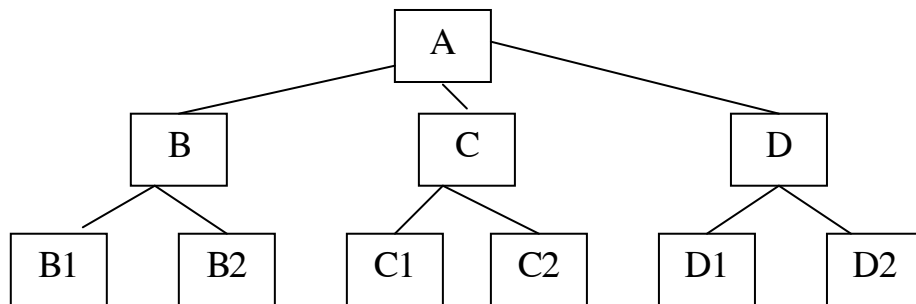
Giao dịch giữa Chủ hàng - Hàng (S - P)

S#	P#	Quality
S1	P1	300
S1	P2	200
S1	P3	400
S2	P1	300
S2	P2	400
S2	P3	200

2- Mô hình phân cấp (Hierarchical Model)

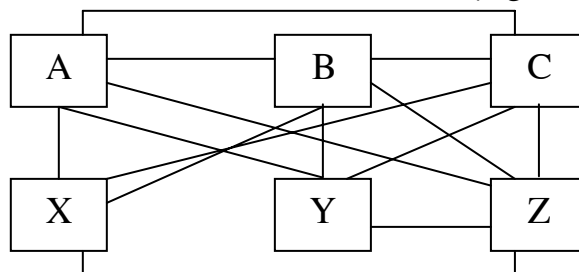
Là mô hình được tổ chức theo cấu trúc phân cấp hình cây theo mối quan hệ cha con giữa các nút trên cây.

Mô hình CSDL này rất khó cài đặt và phức tạp nên ít được sử dụng



3- Mô hình mạng (Network Model)

Là mô hình được tổ chức theo cấu trúc mạng lưới, mối quan hệ đa liên kết rất khó cài đặt và tổ chức nên rất ít được sử dụng



1. 2. LÀM VIỆC VỚI MS ACCESS

1.2.1. Giới thiệu hệ quản trị CSDL

MS – ACCESS là một hệ quản trị CSDL trên môi trường Window trong đó có sẵn các công cụ hữu hiệu và tiện lợi, tự động sản sinh chương trình cho hầu hết các bài toán thường gặp trong quản lý, thống kê, kế toán. Với Access người dùng không phải viết từng câu lệnh cụ thể mà chỉ cần tổ chức dữ liệu và thiết kế các yêu cầu công việc cần giải quyết.

Access cung cấp các đối tượng công cụ phục vụ cho việc quản lý như: Bảng (Table), truy vấn (Query), mẫu biểu (Form), báo biểu (Report), Macro và đơn thể (module).

1.2.2. Vào môi trường Access

Các bước thao tác tiến hành như sau:

- 1- Khởi động Window
- 2- Nháy kép chuột tại biểu tượng Microsoft Access hình chìa khoá trên thanh công cụ nhóm Microsoft Office

Hoặc vào thanh công cụ Start, chọn Program, Microsoft Access

Khi đó sẽ hiện ra màn hình hội thoại của Access có 2 mục trên menu bar là File và Help. Khi đó ta muốn thực hiện tạo mới hay mở một chương trình có thể chọn mục New Database hay Open Database trong thực đơn File.

1.2.3. Chương trình ACCESS

1. Tập chứa chương trình Access

Toàn bộ tập chương trình Access tạo ra có đuôi .MDB nghĩa là tất cả CSDL bao gồm cả dữ liệu và chương trình điều khiển đều được tổ chức chung trong một tập MDB chứ không như các ngôn ngữ lập trình khác.

2. Các đối tượng của Cơ sở dữ liệu

Một CSDL của Access bao gồm các đối tượng sau:

- 1- Bảng (Table): Đó là các bảng (hay File dữ liệu) chứa dữ liệu
- 2- Truy vấn (Query): Là các bảng được tổ chức dưới dạng hỏi/ đáp phục vụ cho việc xử lý và cập nhật dữ liệu.
- 3- Mẫu biểu (Form): Phục vụ cho việc cập nhật và trình bày dữ liệu theo một biểu mẫu nào đó, điều khiển các chức năng xử lý trong chương trình.
- 4- Báo biểu (Report): Phục vụ cho việc tạo lập các bản báo cáo, định dạng và trình bày trang in trong CSDL.
- 5- Tập lệnh (Macro): Tạo ra các hành động bởi các tập lệnh có sẵn của Access

6- Đơn thể (Module): Là các đơn thể chương trình giúp cho việc thiết lập các modul chương trình giúp người lập trình và sử dụng điều khiển các hành động thao tác xử lý trong CSDL.

1.2.4. Tạo một CSDL mới

Các bước thực hiện được tiến hành như sau:

Vào menu File, chọn New Database hoặc kích chuột tại biểu tượng New Database (hình quyển sách màu trắng). Kết quả sẽ hiện ra một màn hình hội thoại.

Để tạo một CSDL nào đó của riêng mình ta gõ tên tệp CSDL vào hộp File name sau đó kích chuột vào nút OK hoặc ấn phím Enter.

Khi đó một CSDL mới đã được thành lập và màn hình sẽ hiện ra một cửa sổ giao diện với tiêu đề là “Database: +[Tên CSDL của bạn]” bao gồm 1 hệ thống menu File, Edit . . . và các công cụ điều khiển Table, Query, Form, Report, Macro, Module.

Nếu bạn không đặt tên cho CSDL thì Access tự động đặt tên CSDL mặc định là DB1.MDB.

1.2.5. Làm việc với một CSDL đã tồn tại

1. Mở một CSDL đã có sẵn.

Để thực hiện mở 1 CSDL đã có sẵn ta thực hiện như sau:

Chọn Menu File, Open Database sau đó chọn tiếp thư mục chứa tệp CSDL cần mở sau đó gõ tên tệp ở hộp File name hoặc kích chuột vào tệp đó rồi chọn OK hoặc nhấn phím Enter.

Khi đó CSDL sẽ được hiện ra và đối tượng Table được chọn ở trạng thái mặc định trong đó có chứa toàn bộ danh sách các bảng của CSDL.

2. Làm việc với các đối tượng của CSDL

Để làm việc với một đối tượng của MS ACCESS ta cần mở cửa sổ chứa đối tượng sau đó chọn đối tượng cần mở trong danh sách các đối tượng : Table, Query, Form, Report, Macro, Module.

1.2.6. Các cửa sổ thường sử dụng khi xây dựng chương trình

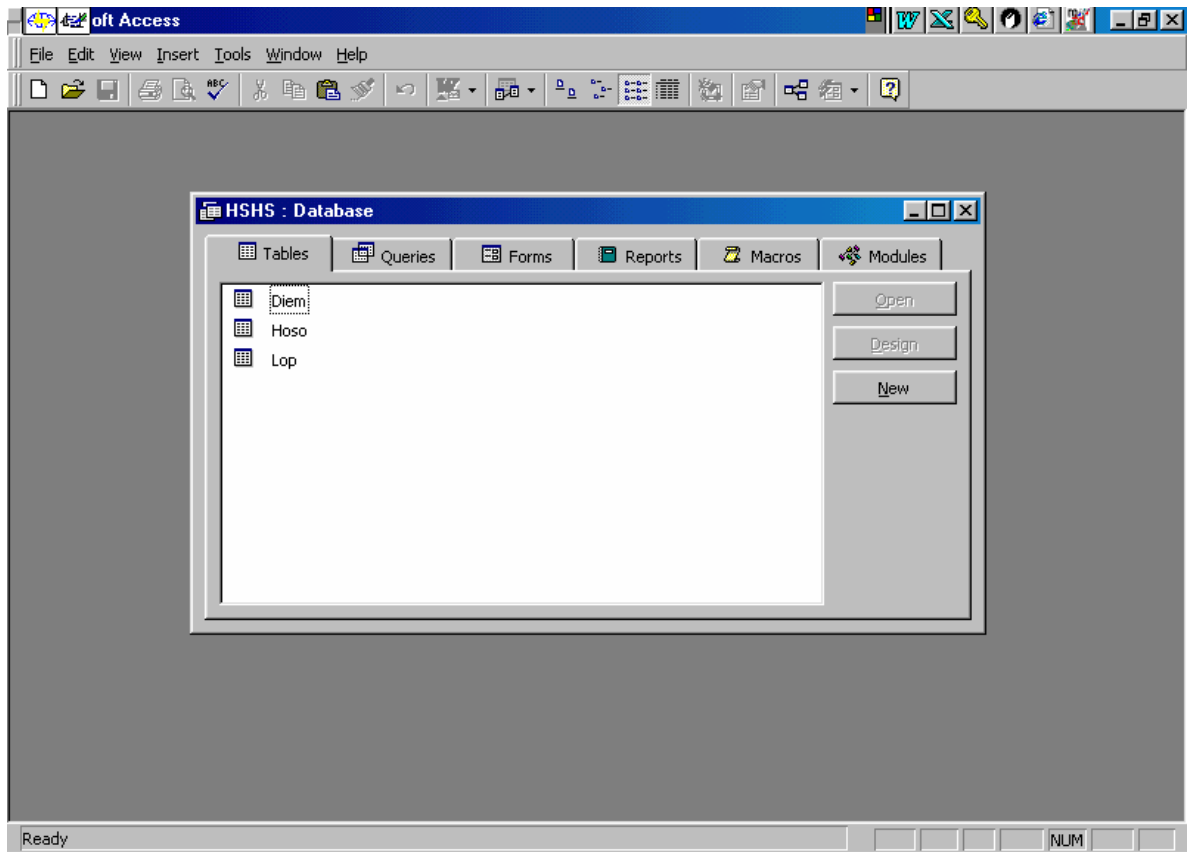
1. Cửa sổ Database

Chứa 6 mục tham khảo tới 6 đối tượng:

- Table: Chứa danh sách các bảng DL của CSDL
- Query: Chứa các truy vấn, hỏi đáp
- Form: Chứa mẫu biểu dùng trình bày hoặc điều khiển cho việc cập nhật và xử lý dữ liệu

Đề cương học phần: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU NGÀNH MAY

- Report: Chứa danh sách các báo biểu phục vụ cho việc làm các bản báo cáo trong CSDL
- Macro: Chứa các tập lệnh thi hành các sự kiện
- Module: Chứa các hàm và thủ tục trong chương trình



2. Cửa sổ Table

Dùng để thiết kế xây dựng cấu trúc bảng CSDL.

3. Cửa sổ Query

Dùng để thiết kế xây dựng các bộ hỏi / đáp cho phép cập nhật hoặc xử lý DL

4. Cửa sổ Form

Dùng để xây dựng mẫu biểu điều khiển chương trình hoặc thiết kế màn hình giao diện trình bày hoặc xử lý cập nhật dữ liệu

5. Cửa sổ Report

Dùng để xây dựng các báo biểu phục vụ cho việc thiết lập báo cáo và định dạng trang in trong CSDL.

6. Cửa sổ Macro

Dùng để xây dựng Macro phục vụ cho việc thi hành chương trình, xử lý các sự kiện.

7. Cửa sổ Module

Dùng để xây dựng các đơn thể, chứa các hàm và thủ tục viết bằng ngôn ngữ Access Basic

1. 3. BẢNG VÀ CÁC THAO TÁC TRÊN BẢNG DỮ LIỆU

1.3.1. Khái niệm bảng

Bảng là nơi chứa dữ liệu của một đối tượng nào đó trong một CSDL. Mỗi bảng bao gồm nhiều trường có các kiểu dữ liệu khác nhau: Text, Date/time, number . . .

Các bảng trong CSDL thường có quan hệ với nhau.

1.3.2. Tạo bảng

Để tạo một bảng CSDL trong môi trường ACCESS có nhiều phương pháp khác nhau. Nếu muốn tạo bảng bằng các công cụ của ACCESS có thể sử dụng các bước sau:

Bước 1: Từ cửa sổ Database chọn mục Table. Kết quả sẽ mở ra cửa sổ Table trong đó chứa danh sách các bảng đã tồn tại và 3 nút lựa chọn New, Open, Design.

Nút New dùng để tạo bảng mới.

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

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

Khi chọn New thì nó sẽ hiện ra một khung cửa sổ hội thoại với các mục (với Access 95 hoặc ACCESS 97):

+ Datasheets View: cho phép hiện lên các trường (Fields) của một bảng dữ liệu cần sử dụng.

+ Table Wizard: Cho phép tạo ra bảng từ một bảng khác đã có sẵn trong CSDL nào đó một cách tự động

+ Design View: Tạo lập một bảng theo cách thiết kế của người sử dụng

+ Link Table: Dùng để liên kết bảng dữ liệu với một bảng khác

+ Import table: Dùng để cập nhật dữ liệu từ một bảng ở trong CSDL hoặc ở CSDL khác sang.

Khi xây dựng bảng theo ý đồ thiết kế của người sử dụng thường dùng chế độ Design View

Bước 2: Chọn Design View khi đó sẽ hiện ra màn hình thiết kế bảng như sau (hình vẽ trang sau):

Cửa sổ Table lúc này được chia ra làm 2 phần:

Phần trên gồm 3 cột Field name, Data Type, Description, dùng để khai báo các trường của bảng, mỗi trường khai báo trên một dòng.

Phần dưới dùng để quy định các thuộc tính cho các trường.

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

Ở phần cửa sổ trên có các cột sau:

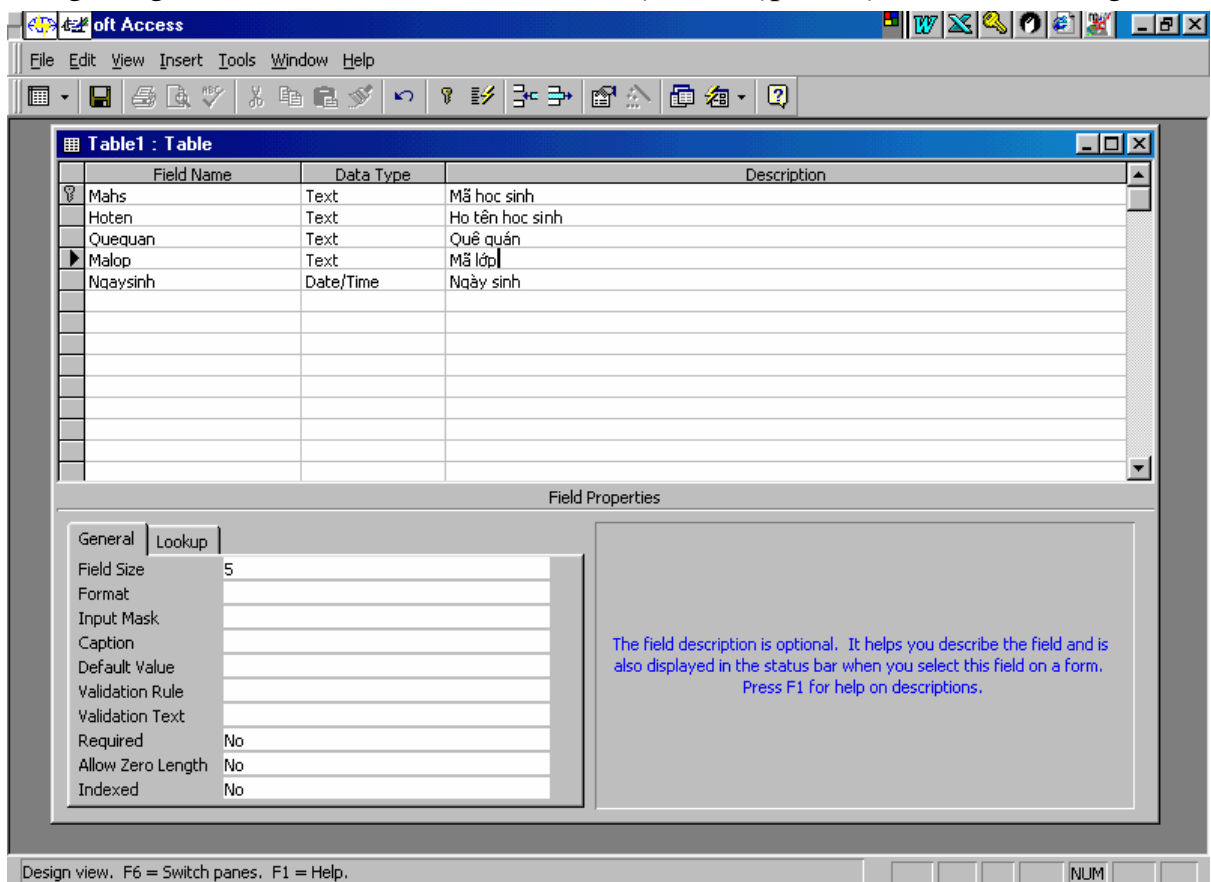
1- Tên trường (Field name): Là một dãy không quá 64 ký tự, bao gồm chữ cái, chữ số, khoảng trống (cho phép) . . .

2- Kiểu dữ liệu (Data Type): Gồm các kiểu sau:

Đề cương học phần: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU NGÀNH MAY

Text	Kiểu ký tự	Độ dài ≤ 255 byte
Memo	Kiểu ký ức	Dài ≤ 64000 byte
Number	Số nguyên, thực	Dài 1, 2, 4, 8 byte
Date/Time	Ngày tháng/ giờ	dài 8 byte
Currency	Tiền tệ	dài 8 byte
Counter	Số	Dài 8 byte
Yes/No	Boolean	1 bit
OLE	Đối tượng nhúng hình ảnh, nhị phân	Dài 1Giga byte

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



Bước 3: Khai báo các trường

- + Đặt tên trường trong cột Field Name
- + Chọn kiểu dữ liệu cần định cho trường đó trong cột Data Type
- + Mô tả trường nếu muốn trong cột Description
- + Để quy định các thuộc tính cho các trường về kiểu, độ rộng, giá trị mặc định ta có thể chọn ở phần cửa sổ Field Properties. Trong đó có các thuộc tính sau:
 Field size: Cho phép đặt kích thước của trường
 Format: Lựa chọn dạng thức chuẩn để hiển thị giá trị của các trường từ danh sách Combo

Input Mask: Các mặt nạ nhập liệu là những chuỗi ký tự, xác định cách hiển thị dữ liệu trong quá trình nhập và hiệu chỉnh

Caption: Đặt tên định danh trường dữ liệu

Default Value: Đặt giá trị mặc định cho trường dữ liệu

Validation Rule: Các quy tắc hợp lệ hoá giúp người dùng thiết lập các quy tắc về tính toàn vẹn dữ liệu với các biểu thức

Validation Text: Văn bản hợp lệ hoá cho phép nhập một giá trị Text, nếu vi phạm biểu thức sẽ đưa ra thông báo lỗi.

Required: Cho phép đưa các yêu cầu bắt buộc tuân theo hay không

Indexed: Đặt chỉ mục cho trường dữ liệu đó hay không . . .

1.3.3. Thay đổi, thiết kế, chỉnh sửa cấu trúc một bảng

+ Xoá một hoặc nhiều trường: Chọn trường cần xoá bấm phím Delete hoặc chọn Delete Row từ Menu Edit.

+ Thay đổi nội dung (tên, kiểu, . . .) của một trường ta chỉ cần dùng bàn phím để thực hiện sự thay đổi cần thiết.

+ Chèn một trường mới: Chọn trường mà trường mới sẽ được chèn vào trước rồi vào Edit chọn Insert Row.

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

1.3.4. Đặt khoá chính cho bảng dữ liệu

+ **Khoá:** Là 1 hoặc nhiều trường xác định duy nhất một bản ghi trong bảng dữ liệu.

+ **Tác dụng:**

Access tự động tạo chỉ mục (Index) trên khoá nhằm tăng tốc độ truy vấn.

Access cho phép kiểm tra dữ liệu trùng lặp khi nhập số liệu nhờ khoá chính.

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

+ **Đặt khoá chính:**

B1: Chọn các trường làm khoá chính

B2: Chọn Edit, Set primary key hoặc kích vào biểu tượng khoá trên thanh công cụ đó.

+ **Xoá khoá chính:**

Muốn xoá khoá chính ra khỏi bảng CSDL thực hiện như sau:

B1: Chọn View, Indexes. Kết quả nhận được trong cửa sổ Indexes có tên trường khoá và cách sắp xếp.

B2: Chọn trường khoá cần xoá rồi nhấn phím Delete

1.3.5. Lưu cấu trúc bảng và đặt tên bảng

Sau khi đã hoàn chỉnh việc thiết kế ta cần ghi lại cấu trúc bảng và đặt tên bảng bằng cách:

Chọn File, Save hoặc kích chuột tại biểu tượng hình đĩa mềm khi đó sẽ hiện ra cửa sổ Save as.

Gõ tên bảng cần lưu giữ vào trong hộp File name sau đó nhấn OK hoặc Enter. Nếu bảng chưa đặt khoá chính thì Access sẽ hiện lên hộp thoại hỏi bạn có muốn thiết lập khoá chính cho bảng hay không. Nếu muốn thì chọn Yes, không thì chọn No. Nếu chọn Yes thì Access tự động đặt trường khoá có tên là ID kiểu Counter.

1.3.6. Thay đổi cấu trúc bảng và nhập dữ liệu vào bảng

1. Thay đổi cấu trúc bảng

Muốn thay đổi cấu trúc của bảng đã tồn tại ta làm như sau:

+ Từ cửa sổ Database chọn mục Tables để mở cửa sổ Tables

+ Trong cửa sổ Tables chọn bảng cần thay đổi cấu trúc, rồi bấm nút Design.

Kết quả thu được cửa sổ thiết kế bảng mà ta muốn thay đổi.

Sau đó tùy theo ý đồ của người thiết kế mà thay đổi cho hợp lý. Sau khi sửa xong phải nhớ ghi lại.

2. Nhập dữ liệu vào bảng

Các bước tiến hành:

+ Từ cửa sổ Database chọn Tables để mở cửa sổ Tables

+ Trong cửa sổ Tables chọn bảng cần nhập dữ liệu, rồi bấm nút Open. Kết quả sẽ mở một cửa sổ dạng như cửa sổ Brow của Foxpro chứa các bản ghi đã nhập.

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

Muốn chọn Font chữ cho bảng dữ liệu ta sử dụng chức năng Font của menu Format.

3. Chuyển đổi giữa hai chế độ nhập liệu và thiết kế bảng

Để chuyển đổi chế độ làm việc nhập liệu và thiết kế bảng ta sử dụng Menu View:

+ Nếu muốn hiện cửa sổ nhập liệu: Chọn Datasheet

+ Muốn hiện cửa sổ thiết kế chọn chức năng Table Design

1.3.7. Lập quan hệ giữa các bảng

+ Access dùng quan hệ để đảm bảo những ràng buộc toàn vẹn giữa các bảng liên quan trong các phép thêm, sửa, xoá mẫu tin.

+ Nguyên tắc quan hệ là chỉ định một hoặc một nhóm trường chứa cùng giá trị trong các mẫu tin có liên quan

Thường đặt quan hệ giữa khoá chính của bảng với 1 trường nào đó của bảng khác, các trường này phải có cùng tên, cùng kiểu, cùng độ rộng.

1. Cách tạo quan hệ

+ Trong cửa sổ Database, chọn Relation Ship từ menu Edit.

+ Nếu chưa thấy cửa sổ Add Table thì chọn Add Table từ menu Relationships.

Kết quả sẽ hiện lên 1 cửa sổ hội thoại có 3 nút tùy chọn: Tables, Queries, Both

Nút Tables hiện các bảng

Nút Queries hiện các truy vấn

Nút Both bao gồm cả các bảng và truy vấn

Chọn các bảng và truy vấn để đưa vào quan hệ trong hộp Table/Query.

Sử dụng các phím CTRL hoặc Shift để chọn nhiều bảng, sau đó bấm nút Add. Sau khi chọn xong, bấm Close để đóng cửa sổ Add Table.

Chọn một trường từ bảng chính (Primary Table) và kéo sang trường tương ứng của bảng quan hệ, rồi bấm chuột tại Create để tạo quan hệ. Khi đó sẽ có đường thẳng nối giữa 2 trường biểu diễn quan hệ vừa tạo.

Sau đó chọn nút Join type để lựa các kiểu quan hệ :

- 1) Only include rows where the joined fields from both tables are equal : Tạo mối quan hệ một – một giữa hai bảng
- 2) Include ALL records from “Bảng chính” and Only those records from “Bảng quan hệ” where the joined fields are equal: Tạo mối quan hệ nhiều - một giữa bảng chính và bảng phụ.
- 3) Include ALL records from “Bảng quan hệ” and only those records from “Bảng chính” where the joined fields are equal: Tạo mối quan hệ một – nhiều giữa bảng chính và bảng phụ

Chọn kiểu quan hệ hợp lý rồi chọn nút OK rồi bấm Create để tạo quan hệ

+ Để xoá quan hệ đã thành lập kích chuột vào đường quan hệ rồi bấm nút Delete.

2. Tính toàn vẹn tham chiếu

Chọn mục Enforce Referential Integrity:

Để thể hiện tính toàn vẹn tham chiếu cần quan tâm đến các điều kiện sau:

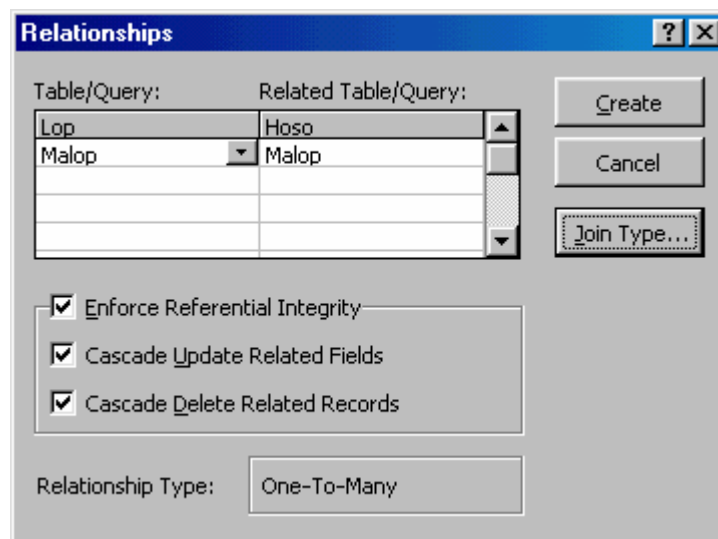
- + Trường của bảng chính là khoá chính
- + Các trường của quan hệ có cùng kiểu dữ liệu và kích thước
- + Cả hai bảng cùng thuộc CSDL

Lúc đó ACCESS sẽ giúp cho người dùng có thể lựa các tùy chọn sau:

+ Cascade Update Related Fields: Tự động cập nhật dữ liệu, khi sửa đổi hoặc cập nhật dữ liệu trong bảng chính thì bảng quan hệ cũng tự động sửa đổi, cập nhật theo.

+ Cascade Deleted Related Fields: Khi xoá 1 bản ghi trong bảng chính thì bản ghi tương ứng trong bảng quan hệ sẽ bị xoá.

Ta có thể hình dung hình ảnh như hình vẽ



3. Xem và điều chỉnh các quan hệ đã có trong Cơ sở dữ liệu

1- Xem các quan hệ đã có

- + Từ menu Edit chọn RelationShips
- + Từ RelationShips chọn Show all để xem tất cả các quan hệ
- + Muốn xem các quan hệ của bảng đó chọn Show Direct trong RelationShips.

2- Chỉnh sửa

- + Chọn quan hệ cần sửa
- + Bấm nút phải chuột
- + Chọn mục Edit Relationship để mở cửa sổ RelationShips
- + Tiến hành chỉnh sửa trong cửa sổ này (Kiểu quan hệ, tính toàn vẹn tham chiếu . . .)

3- Xoá

- + Cho hiện các quan hệ
- + Chọn quan hệ cần xoá (di chuột và bấm)
- + Bấm phím Delete hoặc chọn Delete từ menu Edit

1.3.8. Xem, chỉnh sửa, xoá chỉ mục

Phương pháp này được dùng để xoá khoá chính đã xác lập, để cài đặt khoá chính mới.

-Thao tác:

1. Mở bảng ở chế độ Design View.
2. Chọn Indexes từ menu View. Acces sẽ trình bày cửa sổ Indexes, trong đó thể hiện các trường chỉ mục (kể cả trường khoá chính).
3. Muốn xoá chỉ mục nào (kể cả trường khoá chính), chọn dòng chỉ mục sẽ biến mất trong cửa sổ Indexes.

1. 4. TRUY VẤN VÀ CÁC ỨNG DỤNG CỦA TRUY VẤN

1.4.1. Khái niệm truy vấn

Truy vấn là một công cụ mạnh của ACCESS dùng để : Tổng hợp, sắp xếp và tìm kiếm dữ liệu. Trong các chương 3 và 4 sẽ trình bày cách xây dựng và sử dụng truy vấn.

1.4.2. Các loại truy vấn

Loại thông dụng nhất là Select Query (truy vấn chọn) với các khả năng như :

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

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

Ngoài truy vấn Select Query còn có các loại truy vấn khác là:

- + CrossTab Query: Thể hiện dữ liệu dạng hàng, cột.
- + Action Query: Tạo bảng mới, thêm, sửa, xoá các mẫu tin trong bảng
- + Union Query: Kết hợp các trường tương ứng từ 2 hay nhiều bảng
- + Data – Definition Query: Truy vấn được xây dựng từ một câu lệnh SQL để tạo bảng mới hoặc thay đổi cấu trúc của bảng.

1.4.3. Các bước để thiết kế một truy vấn mới.

Truy vấn là một công cụ mạnh của Access dùng để tổng hợp, sắp xếp và tìm kiếm dữ liệu. Một truy vấn có thể sử dụng các truy vấn khác.

Các bước thiết kế truy vấn như sau:

- B1: Chọn nguồn dữ liệu cho truy vấn mới gồm: các bảng và các truy vấn đã được tạo từ trước.
- B2: Tạo lập quan hệ giữa các bảng, truy vấn nguồn
- B3: Chọn các trường từ các bảng/truy vấn (Table/Query) nguồn để đưa vào truy vấn mới.
- B4: Đưa vào các điều kiện để chọn lọc các mẫu tin thoả mãn các điều kiện đưa vào. Nếu không có các điều kiện chọn lọc thì kết quả truy vấn bao gồm tất cả các bản ghi từ các bảng/truy vấn nguồn.

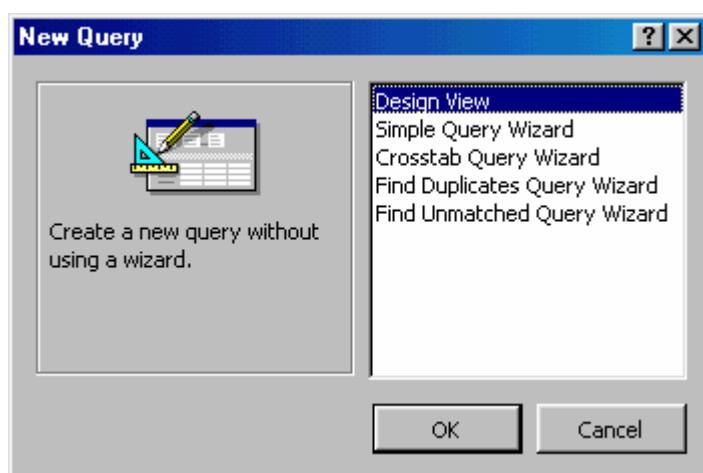
B5: Chọn các trường dùng để sắp xếp các mẫu tin trong Dynaset

B6: Xây dựng các cột (trường) mới từ các trường đã có trong bảng truy vấn nguồn. Các trường mới này là kết quả tính toán của 2 trường sẵn có nhờ việc sử dụng các hàm chuẩn của Access hoặc các hàm tự lập bằng ngôn ngữ Access Basic.

1.4.4. Thiết kế truy vấn

Để thiết kế một truy vấn dựa trên kết quả của một số bảng/ truy vấn nguồn ta thực hiện theo trình tự sau:

+ Trong cửa sổ Database của CSDL chọn mục Query , sau đó bấm New. Khi đó ta sẽ thấy hiện ra một cửa sổ như sau:



Đối với Access 95 và Access 97 có 5 kiểu truy vấn như sau:

+ Design View: Truy vấn thiết kế theo ý của người sử dụng

+ Simple Query Wizard: Truy vấn được tự động thiết kế nhờ công cụ của Access người dùng chỉ việc trả lời các câu hỏi.

+ Crosstab Query Wizard: Truy vấn thể hiện dữ liệu dạng hàng cột được tự động thực hiện nhờ công cụ của Access

+ Find Duplicates Query Wizard:

+ Find Unmatched Query Wizard:

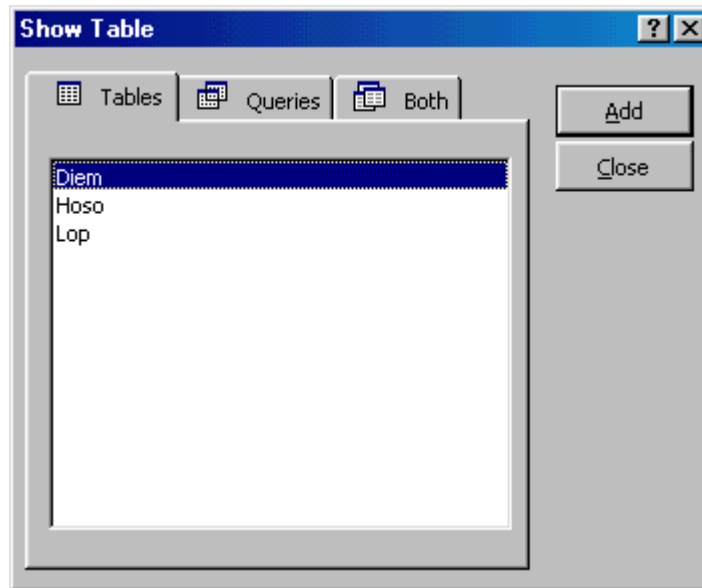
Thông thường để thiết kế theo ý của người dùng thì ta nên chọn cách thiết kế Design View. Khi đó sẽ hiện ra một cửa sổ hội thoại giúp người dùng thiết kế như sau:

Trên cửa sổ có 3 mục chọn :

+ Table : thể hiện danh sách các bảng có trong CSDL giúp cho việc lựa chọn

+ Queries: thể hiện danh sách các truy vấn nguồn đã có sẵn

+ Both: thể hiện cả bảng và truy vấn



Để thực hiện tạo truy vấn mới ta có thể chọn tên các bảng hoặc truy vấn nguồn sau đó click chuột vào nút Add. Khi đã chọn đủ các bảng và truy vấn thì click chuột vào nút Close.

Sau đó tiếp tục tạo ra mối quan hệ giữa các bảng và truy vấn rồi chọn các trường dữ liệu từ bảng và truy vấn nguồn để đưa vào truy vấn cần thiết kể bằng cách kéo tên trường từ các bảng/ truy vấn nguồn từ phần trên đặt xuống dòng Field của phần dưới.

Để chọn nhiều trường ta có thể thực hiện bằng cách đánh dấu các trường cần chọn rồi kéo đồng thời xuống dòng Field .

Để chọn tất cả các trường của một bảng/ truy vấn có thể dùng 2 cách:

+ Cách 1: Chọn tất cả các trường (Bấm và kéo thanh tiêu đề của bảng)

Kéo tất cả các trường xuống dòng Field

+ Cách 2: Kéo dấu * (dòng đầu của bảng) xuống dòng Field.

Khi đó sẽ hiện lên cửa sổ như hình ảnh trang sau:

+ Để thực hiện việc sắp xếp và chèn các trường trong QBE điều chỉnh độ rộng các cột:

- Di chuyển trường:

Đưa chuột lên ngay đầu tên trường sao cho hiện ra dấu mỏ neo ↓ thì bấm chuột để chọn trường.

Đưa chuột vào khối chọn rồi kéo tới vị trí mới

- Chèn thêm trường:

Chọn trường muốn chèn từ danh sách

Kéo trường đó vào cột trong vùng QBE

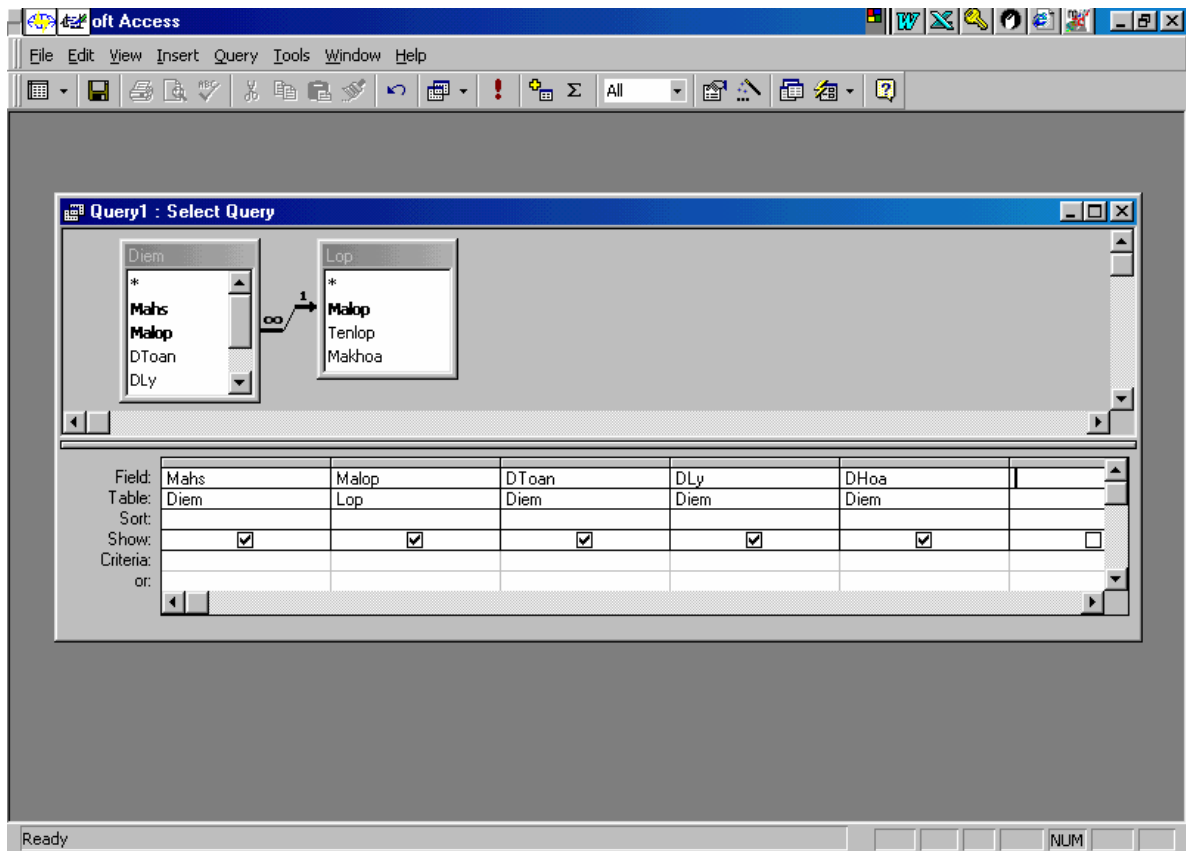
- Xoá trường trong vùng QBE:

B1: Chọn trường cần xoá

B2: Bấm phím Delete hoặc chọn Delete từ menu Edit

- Điều chỉnh độ rộng cột trong vùng QBE:

Có thể dùng chuột di đến đầu cạnh của cột muốn chỉnh cho đến khi hiện dấu mỏ neo ←→ thì kéo tới kích thước mới.



+ Đổi tên trường trong vùng QBE hoặc thêm trường biểu thức

Nếu muốn đưa vào tên một trường mới có ý nghĩa hơn ta có thể gõ tên mới ngay trước tên trường. Đặt dấu : giữa tên mới và tên trường.

Ví dụ: TONGDIEM:[DIEMTOAN]+[DIEMLY]+[DIEMHOA]

+ Sắp xếp: Các bước tiến hành như sau:

B1: Bấm chuột vào ô Sort của trường cần sắp xếp

B2: Chọn chiều Ascending (tăng dần) hoặc Descending (Giảm dần)

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

+ Đặt điều kiện tìm kiếm (tiêu chuẩn lựa chọn):

Gõ trực tiếp biểu thức điều kiện vào ô Criteria của các trường cần đặt điều kiện hoặc có thể kích chuột phải chọn mục Build

+ Ẩn một số trường trong truy vấn: Bấm chuột tại ô trong hàng Show của trường cần ẩn để đưa ô về dạng rỗng

+ Thiết lập các thuộc tính của truy vấn bằng cách chọn mục Properties.

1.4.5. Các dạng hiển thị truy vấn

Bao gồm 3 dạng hiển thị cơ bản như sau:

- + Query Design: Hiển thị dưới dạng thiết kế
- + SQL: Hiển thị câu lệnh SQL tương ứng
- + Datasheet : Hiển thị bảng kết quả của truy vấn

Muốn thay đổi các dạng hiển thị truy vấn ta có thể chọn nút chọn trên thanh công cụ hoặc chọn trên menu View rồi chọn cách hiển thị tương ứng để thi hành.

1.4.6. Hiển thị các mẫu tin không trùng lặp

Để thực hiện việc hiển thị các mẫu tin không trùng lặp thì trong câu lệnh SQL ngay sau mệnh đề Select ta có thể thêm Distinctrow hoặc có thể sử dụng thuộc tính Unique Values trong truy vấn đặt giá trị Yes trong cửa sổ Query Properties.

1.4.7. Chỉnh sửa truy vấn

Một truy vấn trong CSDL Access cho phép thực hiện các công việc:

- + Thêm bớt các bảng/truy vấn nguồn
- + Thêm bớt các trường đưa vào truy vấn
- + Sửa, thêm, xoá các điều kiện tìm kiếm
- + Sửa, thêm, xoá các trường dùng để sắp xếp

Trình tự các bước thao tác như sau:

- B1: Mở truy vấn trong chế độ Design View
- B2: Nếu cửa sổ Add Table đóng thì chọn mục Add Table trong menu Query
- B3: Chọn các bảng/truy vấn cần thêm và bấm nút Add sau đó chọn Close
- B4: Thực hiện các chỉnh sửa cần thiết
- B5: Ghi lại các chỉnh sửa bằng cách: Chọn Save từ menu File hoặc kích chuột vào biểu tượng đĩa mềm.

1.4.8. Xây dựng các điều kiện trong truy vấn

Có thể xử lý các điều kiện trong truy vấn bằng cách sử dụng các hàm số học, hàm logic. . . các hàm chuẩn trong Access Basic để đặt các điều kiện tìm kiếm trong ô Criteria của trường cần đặt điều kiện (*tự đọc các hàm chuẩn để đưa vào điều kiện tìm kiếm sao cho có hiệu quả*).

1.4.9. Tính tổng trong truy vấn

1. Thứ tự các bước làm việc của truy vấn tính tổng

B1: Dựa vào các điều kiện đặt trong các trường (có giá trị Where trên ô Total) để lọc các mẫu tin tham gia phân nhóm và tính toán.

B2: Phân nhóm và sắp xếp các mẫu tin trong từng nhóm theo các trường phân nhóm (Group By)

B3: Thực hiện phép tính theo từng nhóm trên các trường tính toán (có hàm Sum, Avg, . . .) trên ô Total.

B4: Các dòng kết quả có thể được sắp xếp lại theo các trường phân nhóm và các trường tính toán (trong ô Sort)

B5: Nếu có các điều kiện trên các trường phân nhóm và tính toán thì chỉ các dòng thoả mãn điều kiện này mới được hiển thị.

B6: Nếu dùng thuộc tính Top Value của Query thì chỉ 1 số dòng bên trên mới được hiển thị.

2. Thao tác tạo truy vấn tính tổng

1/ Chọn các bảng đưa vào truy vấn

2/ Chọn mục Total của Menu Query. Kết quả xuất hiện dòng Total trên QBE.

3/ Chọn các trường:

3.1. Phân nhóm (Group By trên ô Total)

3.2. Điều kiện, tiêu chuẩn tham gia phân nhóm và tính tổng:

+ Có chữ Where trên ô Total

+ Có biểu thức điều kiện trên ô Criteria

3.3. Tính toán: Chọn một hàm đặt vào ô Total, đổi tên trường cho có ý nghĩa tính toán

3.4. Chọn thứ tự hiển thị: Dùng ô Sort của trường Group By và trường tính toán.

3.5. Định tiêu chuẩn hiển thị kết quả: Đưa điều kiện vào các trường phân nhóm và trường tính toán.

1.4.10. Truy vấn thông số

Đây là loại truy vấn mà giá trị trong các biểu thức điều kiện không cần định trước mà sẽ tự nhập vào mỗi khi truy vấn thực hiện

+ Cách tạo truy vấn thông số

Khi tạo các biểu thức điều kiện, thay cho 1 giá trị cụ thể ta đưa vào một thông báo được bao trong 2 dấu []. Một thông báo như vậy gọi là thông số. Dòng thông báo sẽ xuất hiện như một chỉ dẫn nhập liệu khi thực hiện truy vấn. Đặt thông báo vào trong ô Criteria của truy vấn.

Ví dụ: Đưa vào ô Criteria của trường HOTEN thông báo giúp cho người dùng gõ vào họ tên người cần đưa thông tin ra như sau: Like [Bạn hãy đưa vào họ tên người cần tìm kiếm] như hình ảnh trang sau:

+ Quy định kiểu dữ liệu cho các thông số:

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

B1: Mở truy vấn thông số ở chế độ Design View

B2: Chọn mục Parameters từ menu Query Paramaters

B3: Lần lượt đưa vào tên thông số và chọn kiểu dữ liệu thích hợp.

B4: Chọn nút OK

+ Câu lệnh Paramaters: Để làm cho truy vấn làm việc đúng đắn và ổn định, ta nên dùng câu lệnh Paramaters để định nghĩa các thông số dùng trong truy vấn. Câu lệnh này nên đặt trước câu lệnh SQL của truy vấn và được viết theo mẫu sau:

Parameters [Thông số] Kiểu, . . . , [Thông số] Kiểu;

Ví dụ: Parameters [Quequantim] Text, [Hotentim] Text;

Select Distinctrow [Hocsinh].[mahs], [Hocsinh].[hoten], [Hocsinh].[Quequan]
From [Hocsinh]

Where [quequan] Like [Quequantim] and [hoten] like [Hotentim];

+ Những điều chú ý khi viết thông số:

1/ Tên thông số không thể trùng với tên các trường của bảng/truy vấn nguồn

2/ Thông số không thể bắt đầu bằng khoảng trống

1.4.11. Truy vấn CrossTab

1. Khái niệm và công dụng

- Truy vấn CrossTab được dùng để tóm lược dữ liệu và trình bày kết quả theo dạng cô đọng như một bảng tính, với 1 hình thức rất dễ đọc. Nó thường được dùng để tạo nên số liệu so sánh và tiên đoán hướng phát triển của dữ liệu

- Thực hiện:

+ Trước tiên thực hiện như truy vấn Total: Phân nhóm và tính toán

+ Sau đó trong mỗi nhóm ta lại chia ra các nhóm con, thực hiện phép toán trên mỗi nhóm con và trình bày kết quả theo từng cột.

2. Cấu trúc truy vấn CrossTab

Gồm các trường như truy vấn Total:

+ Các trường phân nhóm (Total: Group By)

+ Các trường tính toán (Total: Sum, AVG, . . .)

+ Các trường điều kiện (Total: Where)

Và thêm 2 trường:

+ Trường phân nhóm theo cột

Ô Total: Group By

Ô CrossTab: Column Heading

+ Trường tính toán trên các nhóm con:

Ô Total: Hàm (Sum, Avg, . . .)

Ô Crosstab: Value

Chú ý: Đối với ô Crosstab thì trường điều kiện để trống, các trường phân nhóm và tính toán nếu để trống thì sẽ không hiển thị, nếu chọn RowHeading sẽ hiển thị giá trị theo hàng.

3. Cách xây dựng truy vấn CrossTab

+ Chọn các bảng/truy vấn nguồn để xây dựng truy vấn mới
+ Chọn mục CrossTab Query trong menu Query kết quả trong QBE xuất hiện thêm 2 hàng là : Total và CrossTab

+ Chọn các trường:

1/ Dùng làm tiêu chuẩn lựa chọn các bản ghi:

Total: Where

CrossTab: Để trống

Criteria: Điều kiện

2/ Phân nhóm và hiển thị theo hàng

Total: Group By

CrossTab: RowHeading

Sort :

Criteria : có thể dùng 2 ô này

3/ Tính theo nhóm và hiển thị theo hàng

Total: Hàm Sum, Avg, . . .

CrossTab: RowHeading

Sort: Có thể dùng

Criteria: Có thể dùng

4/ Một trường dùng để phân nhóm con và hiển thị theo cột

Total: Group By

CrossTab: Column Heading

5/ Một trường dùng để tính toán trên các nhóm con và hiển thị kết quả theo bảng:

Total: Hàm Sum, Count, . . .

CrosTab: Value

1.4.12. Truy vấn dùng câu lệnh SQL

1. Truy vấn hội

+ Công dụng: Dùng để kết nối nhiều bảng có cấu trúc

+ Cách tạo truy vấn hội:

1/ Chọn: Query, New, New Query

2/ Đóng hộp hội thoại Add Table

3/ Chọn mục SQL Specific từ menu Query sau đó chọn Union. Kết quả sẽ mở ra cửa sổ Union.

4/ Soạn thảo câu lệnh SQL trong cửa sổ này

5/ Ghi và đặt tên cho truy vấn

+ Các khả năng của truy vấn hội:

Sắp xếp

Đổi tên trường

Liệt kê các mẫu tin trùng nhau

Ví dụ:

```
Select [HT] as [Hoten], [QQ] as [Quequan]
```

```
From [Hocsinh]
```

```
order by [hoten]
```

```
Union all
```

2. Truy vấn định nghĩa dữ liệu (Data Definition)

1/ Công dụng

Truy vấn định nghĩa dữ liệu có các công dụng sau:

- + Tạo bảng
- + Xoá trường
- + Thêm trường
- + Tạo chỉ mục
- + Xoá chỉ mục

2/ Cách tạo

B1: Chọn Query, New, NewQuery (Access 2.0) hoặc Design View (Access 95 - Access 97)

B2: Đóng hộp hội thoại AddTable

B3: Chọn SQL Specific từ menu Query

B4: Chọn mục Data Definition. Kết quả hiện cửa sổ Data Definition Query

B5: Trong cửa sổ Data Definition Query cho phép sử dụng các câu lệnh sau:

CREATE TABLE (Tạo bảng mới)

ALTER TABLE (Thay đổi cấu trúc của bảng)

DROP TABLE (Xoá bảng)

CREATE INDEX (Xây dựng chỉ mục)

DROP INDEX (Xoá chỉ mục)

B6: Ghi lại câu lệnh đã soạn thảo và đặt tên cho truy vấn

2/ Các ví dụ minh hoạ

VD1: Dùng SQL tạo bảng Hosohs có các trường: Mahs Text (5), Hoten Text (30), Ngaysinh Date Time, Malop Text (5)

Khoá chính được đặt tên là Khoa gồm 2 trường Mahs và Malop

Bảng có 1 chỉ mục có tên là Index1 bao gồm 2 trường Hoten, Ngaysinh

```
Create Table [HosoHS]
```

```
([Mahs] Text (5), [Hoten] Text (30), [Ngaysinh] Date/Time,
```

```
[Malop] Text (5),
```

```
CONSTRAINT Khoa PRIMARY ([Mahs], [Malop]),
```

```
CONSTRAINT index1 UNIQUE ([Hoten], [Ngaysinh]));
```

VD2: Thay đổi cấu trúc bảng

+ Thêm trường Quequan Kiểu Text vào Hosohs

```
ALTER TABLE [HosoHS]
```

- ADD COLUMN [Quequan] Text (30);
- + Xoá trường Hoten ra khỏi bảng
ALTER TABLE [HosoHS]
DROP COLUMN [Hoten];
- VD3: Xoá bảng Hosohs
DROP TABLE [HosoHS];
- VD4: Tạo và xoá chỉ mục
- + Tạo chỉ mục:
CREATE UNIQUE INDEX
Chimuc on [HosoHS] ([Mahs], [Hoten]);
- + Xoá chỉ mục:
DROP INDEX Chimuc on [HosoHS];

1.4.13. Truy vấn con (Sub Query)

1/ Ý nghĩa

Trong truy vấn cho phép xây dựng một truy vấn con tạo bởi một câu lệnh SQL để xây dựng biểu thức điều kiện cho một trường nào đó. Khi đó kết quả của truy vấn con sẽ được sửa trong biểu thức điều kiện

2/ Cách tạo

- + Mở truy vấn trong chế độ Design
- + Trong ô Criteria của một trường nào đó không đưa vào giá trị cụ thể mà đưa vào một câu lệnh SQL trong các dấu ngoặc tròn.

3/ Ví dụ:

Giả sử ta đã có Bảng HosoHS gồm các trường [Mahs], [Malop], [Hoten]. . .
Yêu cầu tìm Hoten các học sinh học cùng lớp T97A.

Ta thực hiện như sau:

B1: Tạo truy vấn gồm 3 trường nói trên

B2: Trong ô Criteria của trường [Hoten] ta đưa vào câu lệnh sau:

(Select [Hoten] from [HosoHS] Where [Malop]='T97A';)

Có thể đặt điều kiện trong cửa sổ Zoom hoặc cửa sổ Build bằng cách bấm nút phải chuột sau đó đưa vào biểu thức điều kiện trong ô Expression Builder

1.4.14. Truy vấn Action

1/ Công dụng, phân loại

- a. Tạo bảng dữ liệu mới(Make table). Kết quả truy vấn(bảng dynaset) được ghi ra đĩa dưới dạng một bảng mới. Bổ sung vào đuôi một bảng đã có(Append query). Kết quả truy vấn được bổ sung vào một bảng .
- b. Xoá các mẫu tin từ một hay nhiều bảng>Delete query).
- c. Chỉnh sửa một dữ liệu trong một hay nhiều bảng(Update query)

2/ Truy vấn tạo bảng

1- Các bước xây dựng truy vấn tạo bảng.

- Tiến hành thiết kế như đối với một truy vấn chọn thông thường.
- Sau đó chọn menu query, make table, Access sẽ yêu cầu đặt tên cho bảng cần tạo trong hộp sau:
- Đặt tên bảng và chọn OK để trở về của sổ thiết kế truy vấn .
- Ghi thiết kế và đặt tên cho truy vấn.
- Thực hiện truy vấn để nhận bảng mới

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

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

- Chọn menu View, Option
- Chọn General.
- Đặt Confirm Action Query: No

3/ Truy vấn bổ sung (Append Query)

Các bước xây dựng truy vấn bổ sung

B1: Tiến hành thiết kế như đối với một truy vấn chọn thông thường

B2: Sau đó chọn menu Query, Append. Access yêu cầu nhập tên bảng cần bổ sung dữ liệu

B3: Nhập tên bảng cần bổ sung dữ liệu

B4: Trong hàng Field: Đưa các trường tương ứng của bảng/ truy vấn nguồn

B5: Trong hàng Append to: Đưa các trường tương ứng của bảng nhận

Chú ý: Nếu bảng/truy vấn nguồn có cùng cấu trúc như bảng nhận thì có thể kéo dấu * của bảng truy vấn nguồn vào hàng Field

4/ Truy vấn xoá

B1: Chọn các bảng/ truy vấn nguồn

B2: Sau đó chọn menu Query, Delete. Kết quả hiện thêm hàng Delete trong nửa dưới của cửa sổ thiết kế truy vấn.

B3: Kéo dấu * của bảng muốn xoá vào ô Field. Khi đó từ From xuất hiện tại ô tương ứng trên dòng Delete. (Chú ý: chỉ xoá được một trong các bảng nguồn)

B4: Kéo các trường để lập điều kiện vào dòng Field. Từ Where xuất hiện trên dòng Delete (Chú ý: Có thể sử dụng trường biểu thức để lập điều kiện. Khi đó cần kéo thêm một số trường dùng để tạo biểu thức)

B5: Ghi thiết kế và đặt tên truy vấn

B6: Thực hiện truy vấn

5/ Truy vấn sửa đổi (Update)

Các bước xây dựng truy vấn sửa đổi

B1: Chọn các bảng/truy vấn nguồn

B2: Chọn menu Query, Update. Kết quả hiện thêm hàng Update to trong nửa dưới của cửa sổ thiết kế truy vấn

B3: Kéo các trường cần sửa và các trường dùng để lập biểu thức điều kiện vào hàng Field

Chú ý: Một trường có thể vừa là trường sửa vừa là trường điều kiện

B4: Đặt các biểu thức vào ô Update to của các trường cần sửa. Giá trị của biểu thức này sẽ là giá trị mới của các trường. Trong biểu thức có thể sử dụng các trường của bảng truy vấn nguồn, các hàm chuẩn của Access và các hàm tự lập.

B5: Đặt các biểu thức điều kiện vào các ô Criteria của các trường điều kiện

B6: Ghi thiết kế và đặt tên truy vấn

B7: Thực hiện truy vấn

CHƯƠNG 2: MẪU BIỂU CẬP NHẬT & TRÌNH BÀY DỮ LIỆU - BÁO BIỂU - MACRO

2.1. MẪU BIỂU – DÙNG MẪU BIỂU ĐỂ CẬP NHẬT DỮ LIỆU

2.1.1. Khái niệm chung về mẫu biểu

Mẫu biểu là một công cụ mạnh của Access dùng để thể hiện và cập nhật dữ liệu cho các bảng, tổ chức giao diện chương trình.

Mẫu biểu gồm các ô điều khiển thuộc các thể loại khác nhau và có công dụng khác nhau. Các Label dùng để hiển thị văn bản, các Object Frame dùng để chứa các hình ảnh, các TextBox dùng để thể hiện giá trị các trường, các biểu thức hoặc gõ từ bàn phím . . .

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

2.1.2. Công dụng của các ô điều khiển

Các ô điều khiển trong mẫu biểu có các công dụng như sau:

1/ Thể hiện dữ liệu

Mẫu biểu có thể gắn với một bảng/truy vấn nguồn hoặc không gắn với một bảng/truy vấn nào cả.

Các ô điều khiển có thể liên kết với các trường của bảng/truy vấn nguồn để thể hiện dữ liệu và cập nhật dữ liệu vào các trường này.

Các ô điều khiển dùng để thể hiện hình ảnh của các trường kiểu OLE, hiển thị kết quả tính toán.

2/ Nhập dữ liệu từ bàn phím

Có thể nhập dữ liệu từ bàn phím thông qua các ô điều khiển dạng TextBox.

3/ Thực hiện hành động

Các nút lệnh trong mẫu biểu có thể gắn với 1 Macro hoặc 1 thủ tục xử lý sự kiện. Sau đó mỗi khi bấm chuột thì Macro hoặc thủ tục gắn với nó sẽ được thực hiện. Có thể dùng nút lệnh để tạo các menu đơn giản cho chương trình.

4/ Tổ chức giao diện chương trình.

Giao diện chương trình có thể được tổ chức dưới dạng một mẫu biểu. Tiêu đề chương trình và các hướng dẫn sử dụng có thể đưa vào các nhãn, trong mẫu biểu có sử dụng các hình ảnh trang trí.

5/ Tổ chức hệ menu Bar

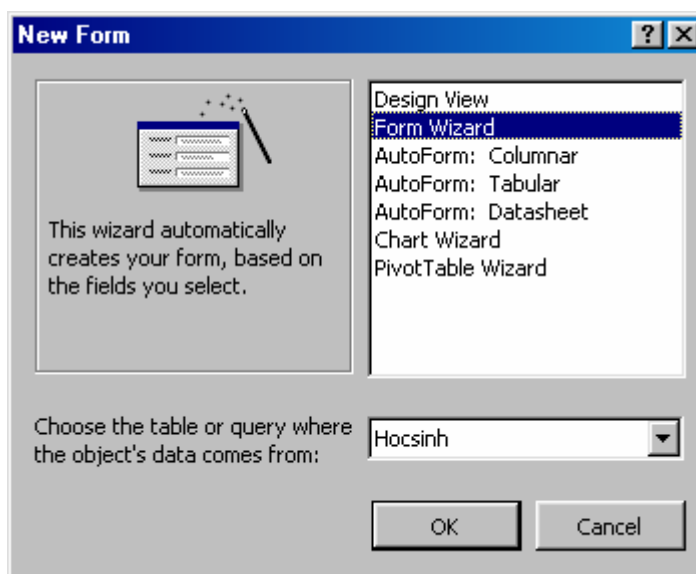
Có thể tổ chức một hệ menu Bar ngay trên mẫu biểu, mỗi menu ngang gồm có các menu thành phần.

2.1.3. Tạo mẫu biểu bằng Wizard

Wizard là một công cụ của Access cho phép người dùng có thể tạo mẫu biểu một cách tự động. Để thực hiện ta tuân theo các bước sau:

1/ Từ cửa sổ Database của CSDL chọn mục Form. Kết quả sẽ mở cửa sổ Forms với 3 nút lựa chọn : New (tạo mẫu biểu mới), Open (thực hiện mở mẫu biểu), Design (mở cửa sổ thiết kế mẫu biểu).

2/ Để tạo mới 1 mẫu biểu ta chọn New. Khi đó sẽ hiện ra cửa sổ NewForm như sau:



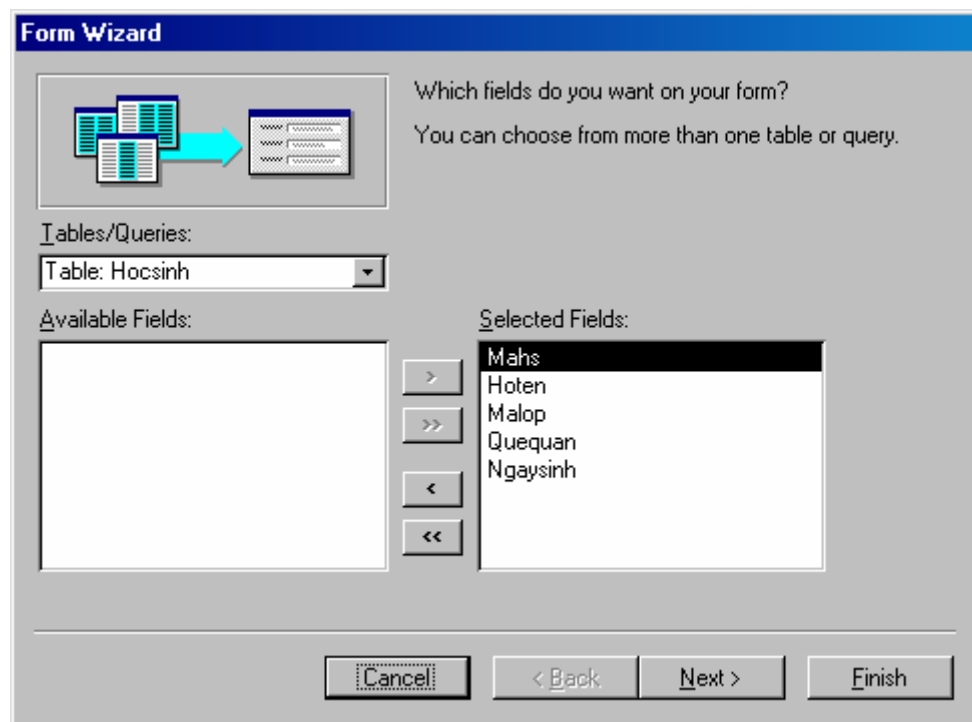
Đối với Access 2.0 thì chỉ có 2 mục là Blank Form và Form Wizard còn Access 95 – 97 thì có các loại :

- + Design View: Hiển thị Form ở dạng thiết kế tùy theo ý thích của người dùng.
- + Form Wizard: Tự động thiết kế Form nhờ công cụ Wizard của Access
- + Auto Form: Columnar : Tự động thiết kế Form nhờ công cụ của Access theo dạng các cột.
- + Auto Form: Tabular: Tự động thiết kế Form nhờ công cụ của Access theo dạng bảng Tabular.
- + Auto Form: Datasheet : Tự động thiết kế Form ở dạng Datasheet
- + Chart Wizard: Thiết kế Form dạng Chart nhờ công cụ Wizard của Access.
- + Pivot Table Wizard: Tự động thiết kế Form dạng bảng Pivot như của Excel.

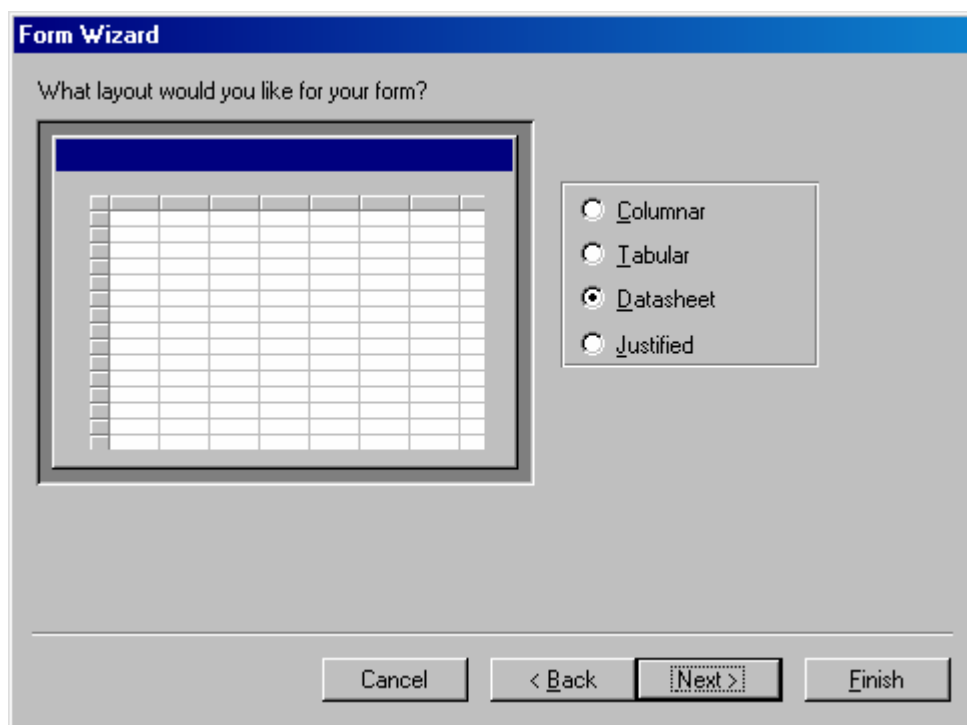
3/ Chọn một bảng/truy vấn nguồn trong ô: Choose the table or query where the object's data comes from (đối với Access 95 –97) hoặc Select A Table/Query (đối với Access 2.0). Nếu không muốn chọn bảng truy vấn nguồn cũng được.

4/ Chọn Form Wizard để sử dụng công cụ Wizard. Kết quả xuất hiện cửa sổ Form Wizards gồm các mục như hình ảnh sau:

Trong ô Table/Queries : Có thể chọn một hoặc nhiều bảng/truy vấn nguồn để lấy dữ liệu. Sau đó ở cửa sổ Selected Fields: Chọn những trường dữ liệu cần thiết đưa vào Form.



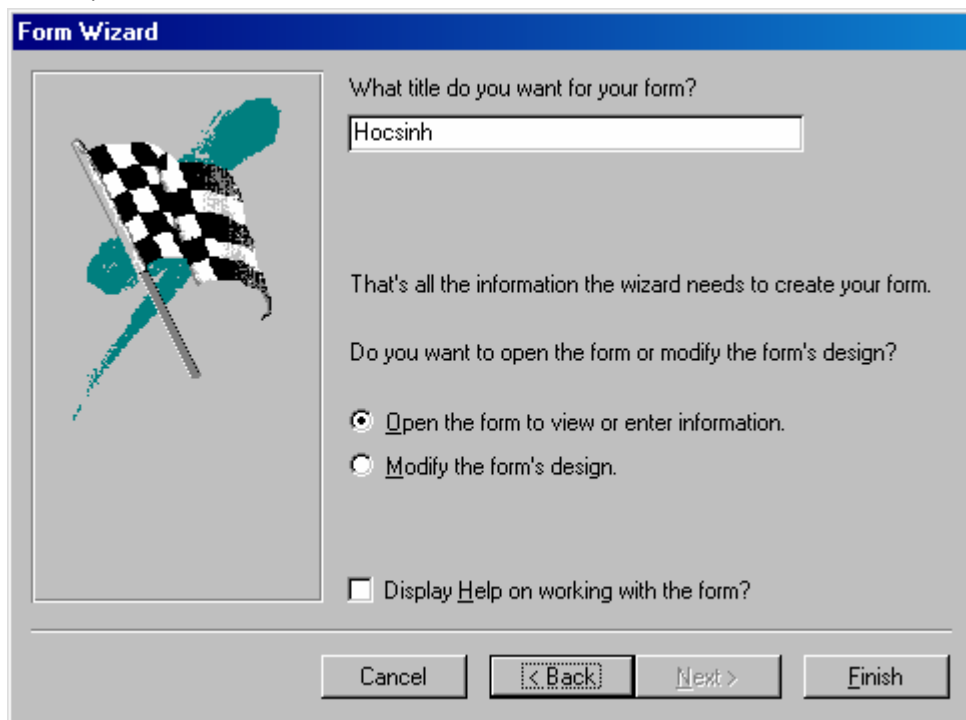
Kích chuột vào nút lệnh Next để chọn cách hiển thị . Khi đó Access sẽ hiển thị ra một cửa sổ như sau:



Lúc đó ta có thể chọn cách hiển thị dữ liệu trên Form ở trong các tùy chọn:
+ Columnar: Hiển thị dữ liệu dạng cột

- + Tabular: Hiển thị dữ liệu dạng bảng
- + Datasheet: Hiển thị dữ liệu dạng bảng Datasheet
- + Justified: Hiển thị dữ liệu dạng canh lề đều 2 bên cho các trường dữ liệu.

Sau khi chọn cách hiển thị xong thì Access tiếp tục hiện ra 1 cửa sổ hội thoại cho phép người dùng chọn cách hiển thị ngay dữ liệu hay quay trở về dạng thiết kế để chỉnh sửa.



- + Open the form to view or enter information: Mở Form xem và nhập dữ liệu
- + Modify the form's design: Sửa đổi lại thiết kế Form.

Lựa chọn 1 trong 2 cách rồi kích chuột vào nút Finish. Khi đó Form đã được hoàn thành.

2.1.4. Mẫu biểu dạng Graph.

1. Đặt vấn đề

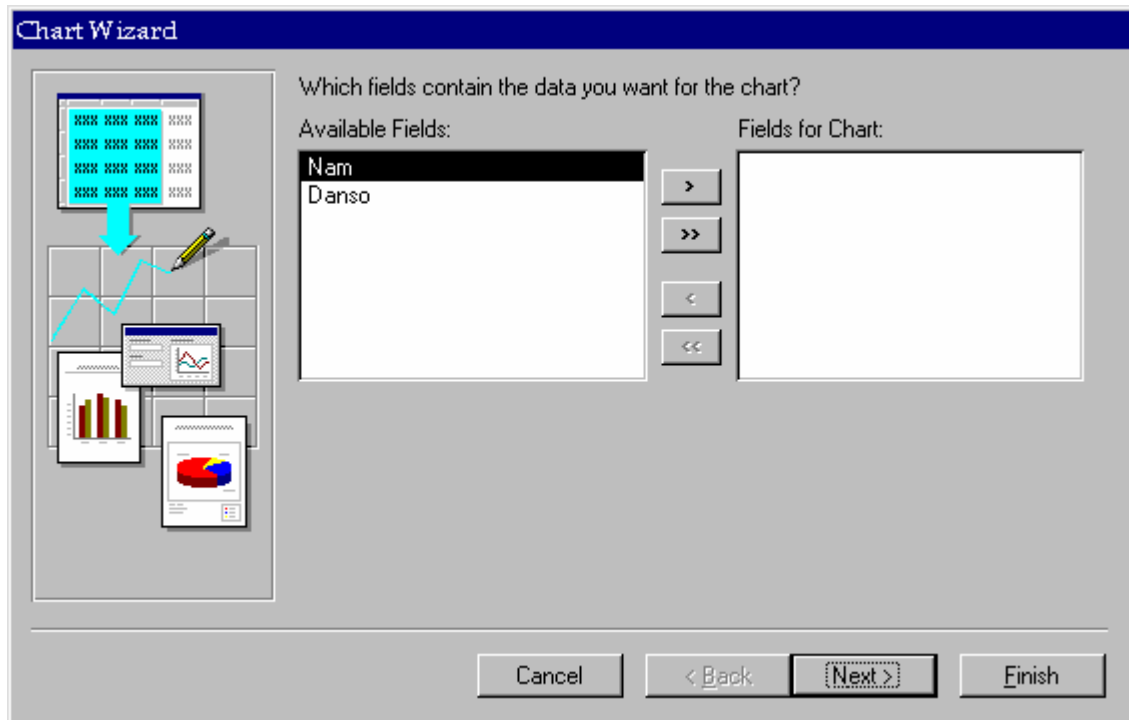
Trong nhiều trường hợp ta muốn biểu diễn dữ liệu của một bảng/truy vấn dưới dạng biểu đồ. Muốn vậy ta sử dụng mẫu biểu dạng Graph do công cụ Wizard cung cấp.

2. Cách thực hiện

1/ Chọn Bảng dữ liệu nguồn

2/ Chọn Form Wizard. Kết quả xuất hiện các cửa sổ liệt kê các dạng mẫu biểu mà Wizard có thể tạo lập.

3/ Chọn dạng Graph. Kết quả hiện cửa sổ như hình ảnh dưới đây:
(Nếu Access 97 có thể chọn Chart Wizard thay cho 2 bước trên)



4/ Chọn các trường để đưa vào mẫu biểu. Sau đó chọn Next.

5/ Chọn trường thích hợp đặt vào trục nằm ngang rồi bấm Next

6/ Có thể chọn 1 trong 3 hàm tính toán sau:

+ Add (Sum) the numbers

+ Average the numbers

+ Count the numbers of records in each category

Sau khi đã chọn các hàm ta tiếp tục bấm Next.

7/ Chọn kiểu biểu đồ cần hiển thị rồi bấm Next.

8/ Soạn tiêu đề cho biểu đồ sau đó bấm nút Finish.

9/ Ghi thiết kế và đặt tên mẫu biểu.

2.1.5. Các loại điều khiển trong mẫu biểu

1. Giới thiệu các loại điều khiển

Khi thiết kế mẫu biểu thường dùng các loại điều khiển sau:

+ Hộp văn bản (Textbox)

+ Nhãn (Label)

+ Hộp lựa chọn (ComboBox)

+ Hộp danh sách (ListBox)

+ Nút lệnh (Command Button)

+ Nhóm lựa chọn (Option Group)

2. Cách dùng

Có thể dùng các ô điều khiển theo 3 cách sau:

+ Bound (buộc vào một trường nào đó)

+ Unbound (Không buộc vào bất kỳ trường nào)

+ Caculated (tính toán)

3. Ý nghĩa của các cách dùng

1/ Điều khiển Bound (buộc vào một trường): Dùng để truy xuất tới các trường đó. Các giá trị cập nhật có thể là hình ảnh, Text, . . .

2/ Điều khiển Unbound (không buộc): Không liên quan đến một trường nào cả. Công dụng của nó:

+ Dùng nhập dữ liệu từ bàn phím

+ Trình bày tiêu đề

+ Trang trí hình, vẽ hộp, đường cong.

3/ Điều khiển Caculated (tính toán): Dùng để thể hiện giá trị của một biểu thức bằng việc thiết lập các biểu thức tính toán trên các trường dữ liệu.

4. Hộp công cụ

Chứa tất cả các công cụ dùng để tạo các ô điều khiển trên mẫu biểu. Toàn bộ các nút công cụ được đặt trong một khung hình chữ nhật có tên là Toolbox.

2.1.6. Điều khiển TextBox

1. Cách sử dụng

TextBox có thể dùng theo 3 cách:

+ Bound

+ Unbound

+ Caculated

2. Tạo TextBox bị buộc (Bound)

Cách 1: Dùng FieldList. Các bước thao tác như sau:

B1: Từ cửa sổ Database chọn Form, New, sau đó chọn Table/Query trong hộp Select a Table/Query làm bảng dữ liệu nguồn.

B2: Chọn DesignView (hay Blank Form trong Access 2.0)

B3: Chọn View, FieldList để mở hộp FieldList chứa danh sách các trường nguồn.



B4: Chọn các trường nguồn kéo vào mẫu biểu. Access sẽ tạo các TextBox bị buộc vào các trường đã lựa chọn.

Khi đó các hộp văn bản bị buộc vào các trường sẽ hiển thị ở các ô tương ứng. Tại đó ta có thể sửa lại tên tiêu đề nhãn đi kèm theo tên ô điều khiển.

Cách 2: Dùng hộp công cụ Toolbox.

B1/ Chọn nút TextBox trên hộp công cụ.

B2/ Di chuột đến vị trí tương ứng trên mẫu biểu sau đó bấm chuột tại đó. Kết quả hiện ra hộp văn bản kiểu Unbound và một nhãn đi kèm bên phải.

B3/ Trong hộp văn bản có chữ Unbound ta sửa bằng một tên trường của bảng/truy vấn nguồn, khi đó hộp văn bản sẽ bị buộc vào trường này.

B4/ Sửa lại nhãn (mặc định Field0) cho có ý nghĩa.

3. Chọn nhãn và hộp văn bản

+ Chọn hộp văn bản: Bấm chuột trên hộp văn bản

+ Chọn nhãn: Bấm chuột trên nhãn

4. Xoá hộp văn bản và nhãn

+ Xoá cả nhãn và hộp văn bản: Chọn hộp văn bản bấm phím Delete

+ Xoá nhãn: Chọn nhãn bấm phím Delete

5. Di chuyển nhãn và hộp văn bản

+ Chọn hộp văn bản hoặc nhãn.

+ Đưa con trỏ đến mép của hộp văn bản hoặc nhãn cho đến khi xuất hiện bàn tay xòe ra thì kéo tới vị trí mới.

6. Phóng to, thu nhỏ

1/ Phóng to thu nhỏ hộp văn bản:

+ Chọn hộp văn bản

+ Đưa chuột tới 1 trong các chấm vuông trên hộp văn bản cho đến khi xuất hiện 2 dấu mũi tên thì kéo tới kích thước mới.

2/ Phóng to, thu nhỏ nhãn:

+ Chọn nhãn

+ Đưa chuột tới các chấm vuông trên nhãn cho đến khi xuất hiện 2 mũi tên thì kéo tới kích thước mới.

3/ Phóng to, thu nhỏ đồng thời:

+ Chọn đồng thời cả hộp văn bản và nhãn

+ Đưa chuột tới 1 trong các chấm vuông nhỏ trên hộp văn bản hoặc nhãn, cho đến khi xuất hiện dấu mũi tên thì kéo tới kích thước mới.

7. Chọn đồng thời các ô điều khiển

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

+ Dùng phím Shift để chọn

+ Bấm chuột tại 1 vị trí trên thước dọc mép trái để chọn các ô điều khiển trên cùng một hàng

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

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

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

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

2.1.7. Tạo các điều khiển khác bằng hộp công cụ

1. Mở hộp công cụ

1. Trong cửa sổ Form chọn mục Toolbox trong menu View.
2. Để hiển thị ý nghĩa của các nút trong hộp công cụ khi di chuột đến đó ta thực hiện như sau:

+ Mở cửa sổ Toolbar sau đó đánh dấu chọn Show Tooltips.

3. Các nút công cụ hay dùng:

+ Label

+ TextBox

+ ComboBox

+ ListBox

+ Command Button

+ Image

+ Sub Form/Sub Report

+ Bound Object Frame

+ Unbound Object Frame

2. Cách tạo các ô điều khiển

1/ Chọn loại điều khiển: Bấm vào nút tương ứng trên hộp công cụ

2/ Di chuột đến vị trí cần đặt điều khiển vào biểu mẫu và bấm chuột

3/ Thực hiện các sửa đổi cần thiết

3. Cố định một loại điều khiển

Để cố định loại điều khiển được lựa chọn bấm nút Toollock (hình chìa khoá). Khi đó Access sẽ cho phép ta cố định một ô điều khiển, cứ mỗi lần bấm chuột tại mẫu biểu thì sẽ có 1 ô điều khiển có loại đang cố định.

4. Dùng TextBox như một điều khiển Unbound hoặc điều khiển tính toán

1/ Dùng hộp công cụ để tạo TextBox thì Access sẽ tạo 1 ô điều khiển mặc định là Unbound. Điều khiển Unbound dùng để: Nhập dữ liệu trong chương trình, hiện kết quả tính toán.

2/ Để biến TextBox thành một điều khiển hiển thị giá trị của biểu thức ta đặt dấu = và biểu thức vào TextBox.

2.1.8. Thuộc tính của điều khiển

1. Mở bảng thuộc tính của một ô điều khiển

Thực hiện như sau:

1/ Chọn điều khiển cần mở bảng thuộc tính

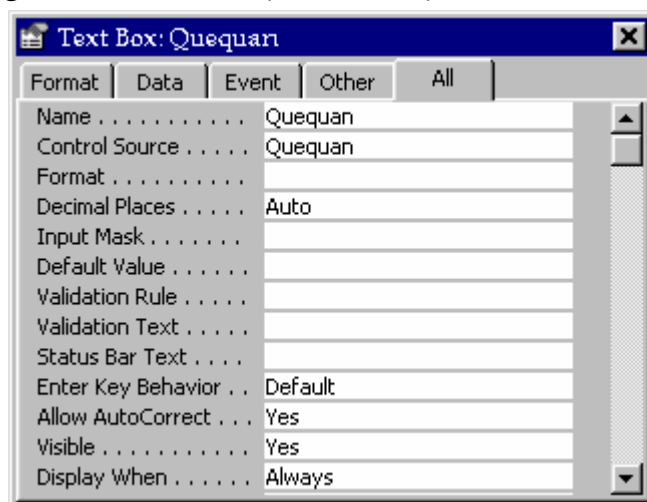
2/ Sau đó có thể thực hiện 1 trong 3 cách:

+ Chọn Properties từ menu View

+ Bấm đúp chuột tại điều khiển

+ Bấm nút phải chuột, chọn mục Properties

Khi đó bảng thuộc tính sẽ hiện ra. Ví dụ như hình sau:



2. Một số thuộc tính hay dùng

1/ Name: Tên ô điều khiển, dùng trong các hàm, thủ tục

2/ Caption: Tiêu đề, thường dùng cho Label

3/ Control source: Nguồn dữ liệu, thường dùng đối với TextBox

4/ Format: Quy định hình thức hiển thị

5/ Default Value: Giá trị mặc định

6/ Validation Rule: Quy tắc hợp lệ dữ liệu

7/ Validation Text: Văn bản hợp lệ

8/ Input Mask: Mặt nạ nhập liệu

9/ Status Bar Text: Hướng dẫn sử dụng điều khiển, hiện dòng trạng thái

10/ Font Color: Màu chữ

11/ Font Name: Tên Font chữ chọn

12/ Font Size: Kích thước của chữ chọn .v.v.

2.1.9. Các thành phần của mẫu biểu

1. Hiển thị các thành phần mẫu biểu

1/ Chọn mục Form Header/Footer từ menu Format để thêm các thành phần: đầu biểu (Form Header) và cuối biểu (Form Footer) vào cuối biểu

2/ Chọn mục Page Header/Footer từ menu Format để thêm các thành phần đầu trang (Page Header) và cuối trang (Page Footer).

2. Năm thành phần của mẫu biểu

Mẫu biểu gồm 5 thành phần được bố trí như sau:

1/ Đầu biểu (Form Header): Dùng để trình bày tiêu đề, hướng dẫn sử dụng các nút lệnh. Có thể cho phép không hiển thị đầu biểu bằng cách đặt thuộc tính Visible: No

2/ Đầu trang (Page Header): Chỉ xuất hiện khi in.

3/ Thân biểu (Detail): Chứa các ô điều khiển buộc vào các trường nguồn .

4/ Cuối trang (Page Footer): Chỉ xuất hiện khi in

5/ Cuối biểu (Form Footer): Sử dụng tương tự Form Header

Có thể thay đổi bề rộng và chiều cao của các thành phần trong mẫu biểu và đặt các thuộc tính.

2.1.10. Một số thao tác khác

1. Các dạng hiển thị mẫu biểu

Bao gồm các dạng sau:

1/ Dạng thiết kế (Design View): Hiển thị mẫu biểu dưới dạng thiết kế

2/ Dạng biểu (Form View): Hiển thị dữ liệu dưới dạng biểu

3/ Dạng bảng (DataSheet View): Hiển thị dữ liệu dưới dạng bảng

4/ Print PreView: Hiển thị dữ liệu như khi in.

2. Các thao tác sau khi thiết kế mẫu biểu

1/ Ghi và đặt tên mẫu biểu: Chọn Save từ menu File

2/ Thực hiện mẫu biểu: Chọn mẫu biểu cần hiển thị rồi bấm nút Open

3/ In Mẫu biểu: Chọn mục Print . . . từ menu File.

2.1.11. Hoàn chỉnh thiết kế mẫu biểu

1. Di chuyển các ô điều khiển

1. Di chuyển 1 ô điều khiển: Chọn ô cần di chuyển, sau đó kéo đến vị trí mới.

2. Di chuyển 1 nhóm điều khiển: Chọn các ô cần di chuyển, sau đó kéo cả nhóm đến một vị trí mới.

2. Các ô điều khiển lồng nhau:

1/ Khái niệm: Ô điều khiển lồng nhau là ô điều khiển mà bên trong của ô điều khiển có chứa các ô điều khiển khác.

2/ Di chuyển các điều khiển lồng nhau:

+ Chọn View, Options để hiện cửa sổ Options chọn tiếp mục Form & Report Design và đặt thuộc tính Move Enclosed Controls thành Yes.

3/ Ẩn hiện các điều khiển bị bao nhau

Để hiện các điều khiển bị bao

+ Chọn điều khiển bị bao

+ Chọn View, Bring to Front

Để ẩn các điều khiển bị bao:

+ Chọn điều khiển bị bao

+ Chọn View, Send to Back

3. Đặt màu cho điều khiển

- 1/ Chọn View, Palete để hiện bảng Palete
- 2/ Chọn điều khiển để trang trí
- 3/ Chọn màu chữ trên dòng Fore Color, chọn màu nền trên dòng back Color, chọn màu viền trên dòng Border Color, chọn dạng nổi, chìm bằng cách dùng các nút ở hàng đầu của bảng Palete

4. Định vị các ô điều khiển

- 1/ Giống thẳng các ô điều khiển

Thao tác như sau:

B1: Chọn các ô điều khiển cần định vị

B2: Chọn Format, Align sau đó:

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

- 2/ Cách đều các ô điều khiển.

+ Cách đều theo hàng:

- 1/ Chọn các điều khiển

- 2/ Chọn Format, Horizontal Spacing, Make Equal

+ Cách đều theo cột:

- 1/ Chọn các điều khiển

- 2/ Chọn Format, Vertical Spacing, Make Equal

5. Thay đổi kích thước

- 1/ Chọn các ô điều khiển cần thay đổi kích thước

- 2/ Chọn Format, Size sau đó chọn kích thước hợp lý

- + Fit: kích thước các ô vừa đủ để hiện nội dung
- + Tallest: các ô có chiều cao bằng chiều cao của ô cao nhất
- + Shortest: các ô có chiều cao bằng chiều cao ô thấp nhất
- + Widest: Các ô có chiều rộng bằng chiều rộng của ô rộng nhất
- + Narrowest: Các ô có chiều rộng bằng ô hẹp nhất

6. Đặt Font chữ cho các ô điều khiển

+ Chọn các ô cần thay đổi Font và cỡ chữ

+ Chọn font và cỡ chữ cần thay đổi

2.1.12. Điều khiển trường Frame và trường OLE

1. Công dụng của điều khiển Frame

- + Điều khiển Frame có 2 cách dùng là Unbound và Bound
- + Điều khiển Frame Unbound dùng để chứa các hình ảnh trang trí không liên quan đến dữ liệu của bảng, truy vấn
- + Điều khiển Bound dùng để buộc dữ liệu kiểu trường OLE chứa hình ảnh trong các bảng CSDL

2. Cách tạo các ô điều khiển Frame buộc vào trường OLE

1/ Trong cửa sổ thiết kế mẫu ta kéo tên trường OLE vào mẫu biểu. Khi đó Access sẽ tạo 1 điều khiển Frame buộc vào trường được kéo và 1 nhãn kèm theo.

2/ Dùng chuột để điều chỉnh kích thước và vị trí của ô điều khiển Frame vừa tạo

3. Đưa ảnh vào các trường OLE

+ Khi đưa 1 hình ảnh vào 1 ô điều khiển Frame thì ảnh sẽ được lưu trữ trong trường OLE tương ứng của bảng/truy vấn nguồn

+ Để đưa 1 File ảnh vào ô điều khiển Frame trước hết cần chọn điều khiển Frame, sau đó dùng chức năng Insert Object của menu Edit

2.1.13. Các loại điều khiển dùng để cập nhật dữ liệu từ bàn phím.

1. Cách dùng

Các điều khiển này có thể dùng theo 2 cách: Unbound và Bound. Khi được buộc vào 1 trường nào đó của bảng/truy vấn nguồn chúng vừa có tác dụng thể hiện dữ liệu và cập nhật dữ liệu cho trường tương ứng. Khi cập nhật dữ liệu người dùng không cần gõ dữ liệu từ bàn phím mà có thể chọn 1 giá trị nào đó trong bảng danh sách các giá trị có sẵn.

2. Cách tạo

Có thể tạo các điều khiển theo 2 cách : dùng Wizard hoặc không dùng Wizard. Trong cửa sổ Form ở dạng thiết kế:

1/ Chọn hoặc không chọn Control Wizard từ menu View

2/ Chọn loại điều khiển trên Toolbox: ListBox, Combo Box, Option Button, ...).

3/ Kéo 1 trường từ hộp Field List vào mẫu biểu (tạo điều khiển Bound). Nếu muốn tạo điều khiển Unbound thì chỉ cần bấm chuột tại 1 vị trí trên Form.

4/ Nếu dùng Wizard thì lần lượt trả lời các câu hỏi, bấm Next, cuối cùng bấm Finish

5/ Nếu không dùng Wizard thì sử dụng các thuộc tính của điều khiển vừa tạo

3. Các loại điều khiển

1/ ListBox:

+ Có 1 danh sách giá trị kèm theo

+ Danh sách luôn được hiển thị

+ Chỉ được chọn trong danh sách, không được nhập dữ liệu từ bàn phím người danh sách

2/ ComboBox:

+ Có 1 danh sách giá trị kèm theo

+ Danh sách chỉ đọc hiển thị khi bấm chuột tại dấu mũi tên đặt bên phải ComboBox

+ Có thể chọn 1 giá trị trong danh sách hoặc nhập 1 giá trị ngoài danh sách.

+ Khi đặt thuộc tính Limit to List: Yes thì Access cho phép chọn từ danh sách như ListBox

+ Khi đặt thuộc tính Limit to List: No thì Access cho phép nhập cả giá trị ngoài danh sách.

3/ Check Box, Option Button, Toggle Button: 3 điều khiển này khác hình thức nhưng cùng công dụng: dùng cho các trường kiểu Yes/No nhận giá trị True (nếu được chọn) hoặc False nếu không được chọn.

4/ Option Group: Thường dùng đối với các trường có giá trị nguyên, miền giá trị từ 1 đến n.

2.1.14. Phân loại ListBox và ComboBox theo nguồn dữ liệu tạo thành các bảng chọn

1. Cấu trúc của ListBox

ListBox và Combo Box có cấu trúc và cách xây dựng hoàn toàn giống nhau. Chúng chỉ khác ở cách thể hiện dữ liệu

Mỗi ListBox và ComboBox có 1 nguồn dữ liệu để xây dựng một bảng chọn. Điểm mấu chốt trong xây dựng ListBox/ComboBox là xác định rõ nguồn dữ liệu của chúng

2. Nguồn dữ liệu của ListBox/ComboBox

Nguồn dữ liệu tạo thành bảng chọn của ListBox/ComboBox bao gồm:

1/ Các bản ghi của bảng/truy vấn

2/ Dãy các giá trị đưa vào khi tạo ListBox/ComboBox (value list)

3/ Tên các trường của 1 bảng/truy vấn (Field List)

4/ Các bản ghi của 1 truy vấn tạo bởi câu lệnh Select

5/ Danh sách tạo từ kết quả của 1 hàm dùng trong Access

2.1.15. Tạo ListBox/ComboBox không cần Wizard

1. Sử dụng các thuộc tính trong bảng thuộc tính để xác định nguồn dữ liệu

Loại (nguồn dữ liệu)	Thuộc tính RowSource Type	Thuộc tính Row Source
Các bản ghi của 1 bảng/truy vấn	Table/Query	Tên bảng/truy vấn
Các bản ghi từ câu lệnh Select	Table/Query	Câu lệnh Select
Danh sách giá trị tự đặt	Value List	Danh sách giá trị phân cách nhau bởi dấu chấm phẩy
Tên các trường của 1 bảng/truy vấn	Field List	Tên bảng/truy vấn
Các giá trị trả về của 1 hàm Access Basic	Tên hàm đó	Để trống

2. Các thuộc tính khác

- + Column Count: Số cột trong danh sách, tính từ trái sang phải
- + Bound Column: Cột được chọn làm nguồn dữ liệu của danh sách
- + Column Width: Độ rộng cột hiển thị
- + Limit to List: No/Yes chọn 1 trong 2 cách có thể nhập hoặc không nhập từ bàn phím
- + List Row: Số hàng của danh sách
- + List Width: tổng bề rộng các cột của danh sách
- + Width: bề rộng điều khiển

3. Sử dụng Value List

Ta có thể đặt giá trị mặc định cho các ô điều khiển dạng này bằng cách chọn các thuộc tính:

- + Row Source Type: Value List
- + Row Source: 1;Giá trị , . . .
- + Column count: số cột

4. Tạo danh sách dùng SQL

Dùng mệnh đề Select trong SQL để thực hiện tạo một danh sách các giá trị trong thuộc tính Row Source:

- + Row Source Type: Table/Query
- + Row Source: SELECT DISTINCTROW các trường DL
From TệpDL . . .

2.1.16. Dùng Wizard tạo ListBox/ComboBox

Các bước tạo ListBox và ComboBox nhờ Wizard:

- 1/ Trước hết phải chọn chức năng Control Wizard trong menu View hoặc bấm vào biểu tượng Control Wizard nếu nó chưa bật
- 2/ Chọn nút ComboBox hoặc ListBox trên hộp công cụ
- 3/ Kéo 1 trường của bảng/truy vấn nguồn vào mẫu biểu. Access sẽ tạo 1 ListBox/ComboBox buộc vào trường này.
- 4/ Lần lượt trả lời các câu hỏi do Wizard đặt ra để xác định nguồn dữ liệu và các đặc trưng khác. Sau khi trả lời 1 câu hỏi bấm Next, cuối cùng bấm Finish.

2.1.17. Dùng điều khiển loại CheckBox, Toggle Button, Option Button

1. Giới thiệu các loại điều khiển

Hộp kiểm tra (CheckBox), nút lựa chọn (Option Button), nút bật tắt (Toggle Button) thường được sử dụng để nhận các giá trị Yes/No. Các điều khiển này chỉ khác nhau về hình thức còn cách sử dụng giống nhau.

- + CheckBox được chọn khi có dấu X trong hộp
- + Option Button: được chọn khi có dấu chấm đen hình hạt đậu bên trong nút hình tròn.
- + Toggle Button: được chọn khi nó như bị nhấn chìm xuống.

2. Cách tạo các điều khiển buộc trường Yes/No.

1/ Chọn 1 trong các nút CheckBox, Option Button, Toggle Button

2/ Kéo 1 trường kiểu Yes/No vào mẫu biểu. Kết quả tạo được 1 điều khiển buộc vào trường vừa kéo và một nhãn đi kèm theo.

2.1.18. Sử dụng nhóm lựa chọn Option Group

1. Công dụng.

Buộc vào các trường kiểu Byte hoặc Integer. Giúp cho các thao tác lựa chọn của người sử dụng dễ dàng hơn vì không phải gõ dữ liệu vào hộp văn bản mà chỉ cần bấm vào một phần tử của nhóm để lựa chọn.

2. Cách tạo.

Trong cửa sổ thiết kế Form:

1/ Không chọn chức năng Control Wizard trong menu View.

2/ Chọn nút Option Group trên ToolBox

3/ Kéo 1 trường có giá trị nguyên vào Form. Tạo 1 điều khiển Option Group buộc vào 1 trường vừa kéo. Điều khiển là 1 hình chữ nhật rỗng ta có thể điều khiển lại

4/ Tạo các nút có lựa chọn bên trong điều khiển nhóm lựa chọn và được xây dựng.

5/ Sửa lại các nhãn của các nút lựa chọn cho hợp với ý nghĩa của bài toán.

2.1.19. Mẫu biểu chính và mẫu biểu phụ (Main/SubForm)

1. Công dụng

Dùng để cập nhập đồng thời cho nhiều bảng, truy vấn:

+ Mẫu biểu chính thể hiện thông tin của một bảng/truy vấn

+ Mẫu biểu phụ thể hiện thông tin của 1 bảng/truy vấn khác có liên quan đến bản ghi đang xét trong mẫu biểu chính.

2. Quan hệ giữa các mẫu biểu chính và phụ

+ Các bảng dữ liệu cần có quan hệ một – nhiều

+ Mẫu biểu chính thể hiện bảng/truy vấn bên một

+ Mẫu biểu phụ thể hiện bảng/truy vấn bên nhiều

3. Giới hạn các mẫu biểu chính và phụ

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

+ Trong mỗi mẫu biểu phụ cấp một cho phép đặt các mẫu biểu phụ khác gọi là mẫu biểu phụ cấp hai.

Access chỉ cho phép tổ chức đến mẫu biểu phụ cấp hai.

2.1.20. Cách tổ chức các mẫu biểu chính và phụ

1. Trình tự thực hiện

- 1/ Tạo một mẫu biểu chính và các mẫu biểu phụ một cách độc lập
- 2/ Sử dụng các thuộc tính của mẫu biểu: View allowed và Default View để quy định dạng trình bày của các mẫu biểu phụ theo bảng thuộc tính sau:

View allowed	Default View	Dạng trình bày của mẫu biểu
Data Sheet	Datasheet	Dạng bảng
Form	Single/Continous Form	Dạng biểu
Both	Datasheet hoặc Single/Continous Form	Cả hai dạng trên

3/ Mở mẫu biểu chính trong chế độ Design.

Bấm phím F11 để xuất hiện đồng thời cửa sổ DataBase

4/ Kéo một mẫu biểu phụ từ cửa sổ DataBase vào trong cửa sổ thiết kế của mẫu biểu chính. Kết quả Access tạo một điều khiển kiểu Subform được đặt mặc định là tên của mẫu biểu phụ được kéo.

5/ Tạo sự liên kết giữa mẫu biểu chính và mẫu biểu phụ

Dùng các thuộc tính điều khiển trong mẫu biểu phụ (SubForm):

+ Link ChildFields:

+ Link MasterFields:

để điền các trường liên kết giữa mẫu biểu chính và mẫu biểu phụ

+ Trong dòng thuộc tính Link MasterFields ta ghi lại các trường liên kết của mẫu biểu chính

+ Trong dòng Link ChildFields ta ghi lại các trường liên kết của mẫu biểu phụ

2.1.21. Tạo mẫu biểu phụ cấp hai

Trong trường hợp mẫu biểu chính đã có mẫu biểu phụ cấp một. Để tạo một mẫu biểu phụ cấp 2 thể hiện thông tin liên quan đến bản ghi hiện hành của mẫu biểu phụ cấp 1 ta thực hiện như sau:

1/ Tạo 1 hộp văn bản tính toán (Calculated) để thể hiện nội dung của một ô điều khiển của mẫu biểu phụ cấp một. Hộp văn bản này sẽ được dùng để liên kết với mẫu biểu phụ cấp hai. Đặt tên thuộc tính name cho mẫu biểu phụ này. Tên này sẽ dùng trong bước 3 để tạo sự liên kết giữa mẫu biểu chính và mẫu biểu phụ cấp 2.

2/ Kéo mẫu biểu dùng làm mẫu biểu phụ cấp 2 vào mẫu biểu chính. Bằng cách này tạo được một điều khiển kiểu SubForm buộc vào mẫu biểu phụ cấp 2.

3/ Tạo mối liên kết giữa mẫu biểu phụ cấp 1 và mẫu biểu phụ cấp 2 bằng cách sử dụng các thuộc tính Link MasterFields và Link ChildFields của điều khiển SubForm như sau:

+ Trong dòng Link MasterFields ghi tên của ô điều khiển tính toán dùng để liên kết

+ Trong dòng Link ChildFields: ghi tên của trường liên kết của mẫu biểu phụ cấp hai

2. 2. BÁO BIỂU – SỬ DỤNG BÁO BIỂU ĐỂ TRÌNH BÀY DỮ LIỆU

2.2.1. Khái niệm về báo biểu, so sánh báo biểu với mẫu biểu

1. Công dụng của báo biểu

Báo biểu là phương pháp nhanh chóng và hữu hiệu để tổ chức in dữ liệu từ các bảng, truy vấn dưới các dạng bảng, biểu.

Báo biểu có các khả năng in ấn như sau:

- 1/ In dưới dạng biểu như hoá đơn, thẻ dự thi, phiếu báo điểm, giấy mời, . . .
- 2/ In dưới dạng bảng như bảng lương, danh sách cán bộ, . . .
- 3/ Sắp xếp và in theo từng nhóm, mỗi nhóm có thể bắt đầu in từ trang mới, như in danh sách thí sinh theo phòng thi, . . .
- 4/ Sắp xếp phân nhóm và thực hiện các phép tính (tổng, max, min, trung bình cộng, đếm, . . .) trên mỗi nhóm
- 5/ In dữ liệu từ nhiều bảng có liên quan trên cùng một trang.

2. Sự giống và khác nhau giữa báo biểu và mẫu biểu

Giữa báo biểu có rất nhiều sự giống nhau như:

- + Cả báo biểu và mẫu biểu có thể có hoặc không có dữ liệu, nếu có thì dữ liệu cũng là một bảng/truy vấn
- + Cả báo biểu và mẫu biểu đều có thuộc tính RecordSource và có thể sử dụng thuộc tính này để xây dựng một truy vấn nguồn cho mẫu biểu và báo biểu
- + Cấu trúc của báo biểu cũng giống như mẫu biểu gồm các ô điều khiển.
- + Trên báo biểu có những sự kiện giống như trên mẫu biểu có thể gắn các Macro và các thủ tục xử lý sự kiện
- + Có thể thiết kế báo biểu bằng cách dùng 1 hộp công cụ như thiết kế mẫu biểu, cách sử dụng giống nhau.
- + Có thể sử dụng công cụ Wizard để thiết kế báo biểu.

Các điểm khác của báo biểu:

- 1/ Báo biểu cũng có 5 thành phần như mẫu biểu: Đầu báo biểu, thân trang, thân báo biểu, cuối trang, cuối báo biểu. Ngoài ra còn có khả năng phân nhóm nên có thêm các thành phần: đầu nhóm, cuối nhóm, trong mỗi nhóm lại có các nhóm con, do vậy lại có đầu nhóm con và cuối nhóm con.
- 2/ Ô điều khiển TextBox của báo biểu có thuộc tính Running Sum cho phép tạo thêm cột số thứ tự trên toàn báo biểu, hoặc trên từng nhóm của báo biểu.
- 3/ Cửa sổ Report có 3 mục : New, Preview, Design khác với cửa sổ Form có: New, Open, Design.
- 4/ Chế độ hiển thị báo biểu có 3 chế độ:
 - + Design View: Dạng thiết kế báo biểu, dùng để chỉnh sửa lại thiết kế của 1 báo biểu
 - + Samle Preview: Xem thử kết quả của báo biểu, kiểm tra cách sắp đặt các điều khiển, font và cỡ chữ. Chỉ hiện ra 1 phần báo biểu để xem mẫu

+ Print Preview: Xem thử và kiểm tra toàn bộ nội dung, cách trình bày của báo biểu trước khi đưa ra máy in.

2.2.2. Xây dựng báo biểu bằng cách sử dụng hộp công cụ

1. Các vấn đề thường gặp khi xây dựng báo biểu.

Khi xây dựng báo biểu thường xuyên gặp các vấn đề sau đây:

- 1/ Xây dựng đầu biểu
- 2/ Xây dựng thân biểu và chỉnh lý thân biểu để khớp với đầu biểu
- 3/ Tạo nền mờ cho 1 đoạn văn bản cần nhấn mạnh
- 4/ Chọn Font chữ, cỡ chữ, kiểu in, căn lề
- 5/ Đánh số thứ tự cho báo biểu
- 6/ Ngắt trang theo yêu cầu
- 7/ Đánh số trang

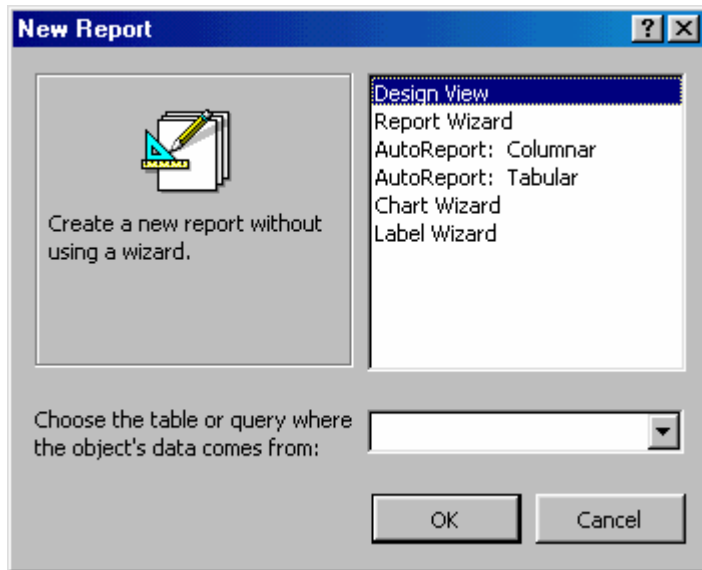
2. Các công cụ thường sử dụng

Khi xây dựng báo biểu ta thường xuyên sử dụng các công cụ sau:

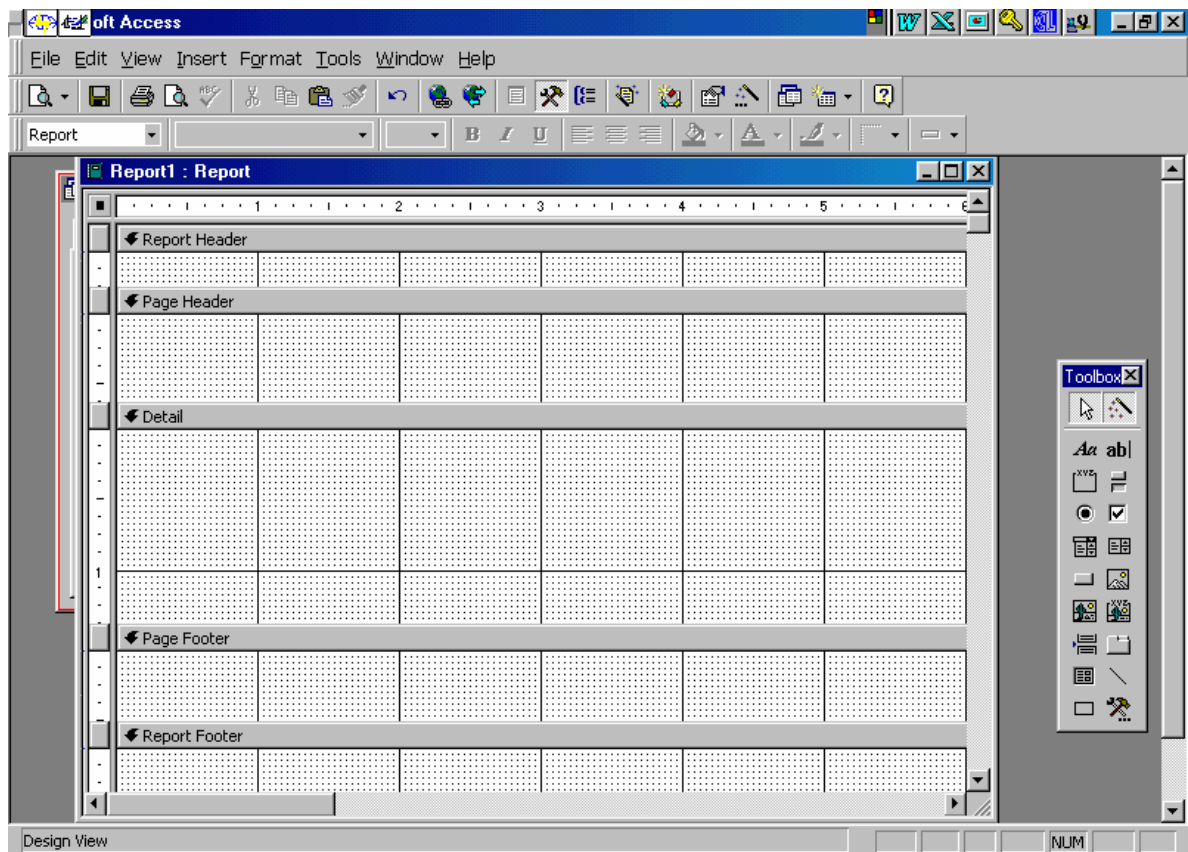
- 1/ Điều khiển Line/Rectangle để kẻ đường thẳng và các hình chữ nhật
- 2/ Chức năng Duplicate của menu Edit dùng để nhân bản một nhóm điều khiển được chọn
- 3/ Chức năng Align và Size của menu Format để chỉnh lý các ô điều khiển tạo nên đầu biểu và thân biểu
- 4/ Sử dụng tổ hợp phím CTRL+ các phím mũi tên để di chuyển cả khối điều khiển được chọn theo các bước ngắn
- 5/ Sử dụng hộp Palette để: chọn kiểu đường kẻ tại dòng đầu tiên, tạo viền cho các ô điều khiển, tạo nền mờ cho các ô điều khiển
- 6/ Chọn Font chữ, kiểu chữ, cỡ chữ, căn lề
- 7/ Để tạo thêm cột số thứ tự ta làm như sau:
 - + Tạo 1 điều khiển TextBox kiểu Unbound
 - + Gán số 1 cho điều khiển bằng cách đặt =1 vào thuộc tính Control Source
 - + Đặt thuộc tính Running Sum là:
 - OverGroup nếu muốn đánh số thứ tự cho từng nhóm
 - Over All nếu muốn đánh số thứ tự cho toàn báo biểu.
- 8/ Sử dụng điều khiển Page Break để ngắt trang
- 9/ Để in số trang ta dùng 1 hộp văn bản kiểu tính toán trong đó ghi công thức = Page.

3. Các bước xây dựng báo biểu

- 1/ Từ cửa sổ Database, chọn mục Report, rồi chọn New. Kết quả xuất hiện cửa sổ Report như hình ảnh sau:
- 2/ Chọn một bảng hoặc truy vấn nguồn trong hộp Select A Table/Query đối với Access 2.0 hoặc Choose 's the table or query where object 's data comes from. Có thể bỏ qua việc chọn bảng/truy vấn nếu không cần sử dụng nguồn dữ liệu hoặc chọn bảng sau này nhờ thuộc tính Record Source. Kết quả nhận được cửa



số thiết kế báo biểu có dạng:



3/ Sử dụng hộp công cụ để tạo các điều khiển trên báo biểu

4/ Sử dụng các công cụ để kẻ, trang trí cho báo biểu

5/ Sau khi hoàn tất thiết kế, ghi lại và đặt tên cho báo biểu

8.2.4. Thực hiện việc in báo biểu

1/ Để in kết quả của báo biểu ra máy in ta làm như sau:

+ Chọn báo biểu cần thực hiện in từ cửa sổ Database

+ Chọn chức năng Print từ menu File, Print

2/ Để in các thông tin về cấu trúc của báo biểu, ta thường dùng chức năng Print Definition trong menu File.

2.2.3. Sắp xếp dữ liệu và tập hợp dữ liệu theo nhóm.

1. Sắp xếp dữ liệu

+ Khi xây dựng báo biểu ta có thể sắp xếp để kết quả in ra theo một trình tự nào đó.

+ Các khả năng của sắp xếp:

1/ Có thể sắp xếp theo trường

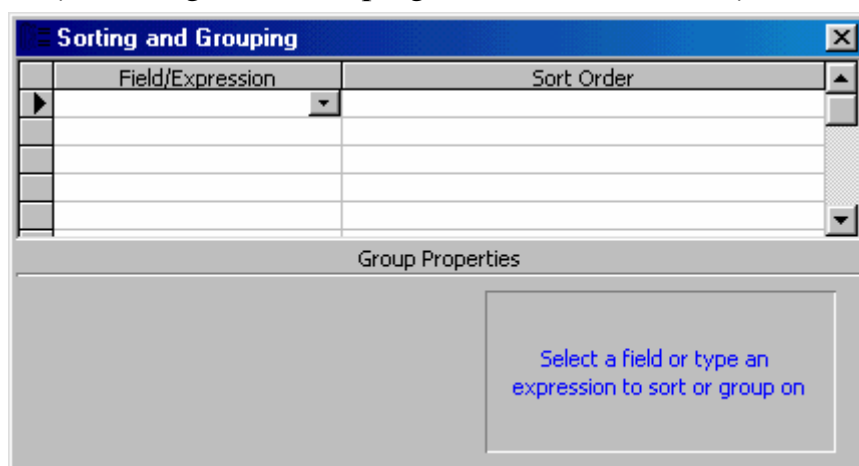
2/ Có thể sắp xếp theo một biểu thức chứa các trường

3/ Có thể sắp xếp tối đa trên 10 trường hoặc biểu thức

+ Muốn sắp xếp dữ liệu trên báo biểu ta làm như sau:

1/ Mở báo biểu trong chế độ Design View

2/ Chọn mục Sorting And Grouping từ menu View để hiện cửa sổ :



3/ Đưa các trường và biểu thức dùng để sắp xếp vào cột Field/Expression

4/ Chọn thứ tự sắp xếp

2. Phân nhóm dữ liệu

1/ Có thể phân nhóm trên bất kỳ trường hay biểu thức nào đã được sắp thứ tự trong cột Fields/Expression

2/ Cách nhóm dữ liệu theo trường phân nhóm

Cách nhóm dữ liệu được quy định bằng các thuộc tính Group on và Group Interval của trường phân nhóm trong cửa sổ Sorting And Grouping.

3. Đầu nhóm và cuối nhóm

1/ Tạo đầu nhóm và cuối nhóm trên báo biểu: Cách thực hiện như sau:

+ Mở cửa sổ Sorting And Grouping

+ Chọn trường hay biểu thức phân nhóm

+ Đặt thuộc tính Group Header là Yes để thêm phần đầu nhóm hoặc No để huỷ phần đầu nhóm.

+ Đặt thuộc tính Group Footer là Yes để thêm hoặc huỷ phần cuối nhóm.

2/ Đặt các ô điều khiển trên phần đầu/ cuối nhóm.

3/ Tổng hợp dữ liệu trên từng nhóm

Tại đầu/cuối nhóm ta có thể sử dụng 1 số hàm chuẩn của Access để thực hiện việc tổng hợp dữ liệu của mỗi nhóm.

4/ In theo nhóm

Thuộc tính Keep Together trong cửa sổ Sorting And Grouping cho 2 khả năng in dữ liệu của nhóm như sau:

+ Nếu: Keep Together : Whole Group

thì nhóm sẽ bắt đầu in từ trang mới nếu phần còn lại của trang không chứa nổi tất cả các bản ghi của nhóm.

+ Nếu: Keep Together: With First Detail

thì trên mỗi trang, tiêu đề nhóm sẽ được in kèm với ít nhất một bản ghi

2.2.4. Nhóm dữ liệu theo trường phân nhóm.

1. Cách phân nhóm đối với trường/biểu thức kiểu số

Có 2 cách phân nhóm: Theo giá trị và theo miền giá trị. Để chọn cách phân nhóm ta sử dụng thuộc tính Group On

+ Phân nhóm theo giá trị: Tại thuộc tính Group On: Each value

Khi đó các bản ghi được sắp xếp theo thứ tự tăng hoặc giảm của trường/biểu thức phân nhóm, sau đó các bản ghi có cùng giá trị trên trường/biểu thức phân nhóm sẽ được đưa vào một nhóm.

+ Phân nhóm theo miền giá trị: Đặt thuộc tính Group On : Interval

Đặt thuộc tính Group Interval: là 1 giá trị

Khi đó miền phân nhóm và các đoạn có độ dài bằng giá trị đưa vào thuộc tính Group Interval và mốc là giá trị 0.

VD: Group On: Interval

Group Interval: 5 thì miền phân nhóm có độ dài 5 và mốc giá trị 0 gồm các đoạn:

. . . [-10, -6], [-5, -1], [0, 4], [5, 9], [10, 14]. . .

Các bản ghi có giá trị trường/biểu thức phân nhóm rơi vào cùng 1 đoạn sẽ được đưa vào một nhóm.

2. Cách phân nhóm đối với trường biểu thức kiểu Date/time.

Để thực hiện việc phân nhóm đối với các trường biểu thức kiểu Date/time có 2 cách:

1/ Phân nhóm theo giá trị

Đặt thuộc tính Group On: Each Value

Khi đó các bản ghi được sắp xếp theo thứ tự tăng hoặc giảm của trường biểu thức phân nhóm, sau đó các bản ghi có cùng giá trị trên trường/biểu thức phân nhóm sẽ được đưa vào cùng một nhóm.

2/ Phân nhóm theo miền giá trị

Miền giá trị có thể tính theo 1 trong các đơn vị: Year (năm), Qtr (quý), Month (tháng), Week (tuần), Day (ngày), Hour (giờ), . . .

Để phân nhóm theo miền giá trị ta đặt:

+ Thuộc tính Group On là 1 trong các đơn vị trên

+ Thuộc tính Group Interval là 1 biểu thức số. Khi đó miền phân nhóm được xác định bằng 2 thuộc tính trên

VD: Group On: Month

Group Interval: 6 thì miền phân nhóm là các khoảng thời gian 6 tháng một, bắt đầu tính từ tháng thứ nhất trong năm

3. Cách phân nhóm đối với trường biểu thức kiểu Text

Có 2 cách phân nhóm: theo nhóm giá trị và theo ký tự đầu

1/ Phân nhóm theo giá trị

Đặt thuộc tính Group On: Each Value

Khi đó các bản ghi được sắp xếp theo thứ tự tăng hoặc giảm của trường biểu thức phân nhóm, sau đó các bản ghi có cùng giá trị trên các trường/biểu thức phân nhóm sẽ được đưa vào một nhóm.

2/ Phân nhóm theo các ký tự đầu

Đặt thuộc tính:

+ Group On là Prefix Character

+ Group Interval: là 1 giá trị nguyên n

Khi đó các bản ghi trùng nhau trên n ký tự đầu của trường/biểu thức phân nhóm được đưa vào 1 nhóm.

2.2.5. Tổng hợp dữ liệu trên báo biểu

1. Cách tổng hợp dữ liệu trên báo biểu

1/ Các hàm tổng hợp dữ liệu trên báo biểu

Để tổng hợp dữ liệu trên các nhóm và trên toàn bộ báo biểu ta có thể sử dụng các hàm sau:

+ Sum : tính tổng

+ AVG: Tính giá trị trung bình

+ Min: Tìm giá trị nhỏ nhất

+ Max: Tìm giá trị lớn nhất

+ Count: Đếm số bản ghi

+ First: Cho bản ghi đầu tiên

+ Last: Cho bản ghi cuối cùng

2/ Tổng hợp dữ liệu trên mỗi nhóm

Để tổng hợp dữ liệu của các bản ghi trong nhóm ta tạo các điều khiển Unbound tại đầu hoặc cuối nhóm và đặt vào ô điều khiển các hàm nói trên

3/ Tổng hợp dữ liệu trên toàn báo biểu

Để tổng hợp dữ liệu trên các bản ghi của toàn bộ báo biểu, ta cũng làm theo cách trên, nhưng các điều khiển được đặt tại đầu hoặc cuối báo biểu.

4/ So sánh dữ liệu tổng hợp trên các phần của báo biểu

Khi xây dựng báo biểu thường phải so sánh, đối chiếu số liệu tổng hợp giữa các phần với nhau. Để so sánh dữ liệu tổng hợp trên các phần của báo biểu ta làm như sau:

1- Tạo các ô điều khiển tại đầu hoặc cuối mỗi phần để nhận dữ liệu tổng hợp trên các phần. Dùng thuộc tính Name đặt tên cho các ô điều khiển

2- Sử dụng các ô điều khiển này để thực hiện các phép so sánh dữ liệu tổng hợp trên các phần khác nhau của báo biểu.

3. Không in 1 số ô điều khiển của báo biểu

+ Nếu đặt thuộc tính Visible: No thì nội dung của nó không được in ra nhưng trên giấy vẫn có 1 khoảng trống đúng bằng kích thước của điều khiển

+ Nếu đặt thuộc tính Height: 0 thì nội dung của nó sẽ hoàn toàn bị bỏ qua trên kết quả in ra của báo biểu

2.2.6. Báo biểu chính và báo biểu phụ

1. Cách tổ chức các báo biểu chính và báo biểu phụ

Trình tự thực hiện như sau:

B1: Tạo 1 báo biểu chính và các báo biểu phụ một cách độc lập

B2: Mở báo biểu chính trong chế độ Design. Bấm phím F11 để đồng thời xuất hiện cửa sổ Database

B3: Kéo 1 báo biểu phụ từ cửa sổ Database vào trong cửa sổ thiết kế của báo biểu chính. Kết quả Access tạo ra 1 điều khiển kiểu SubReport buộc vào báo biểu phụ và 1 nhãn đi kèm

B4: Tạo sự liên kết giữa báo biểu chính và báo biểu phụ. Dùng các thuộc tính: Link Child Fields và Link MasterFields để điền vào các trường liên kết giữa báo biểu chính và báo biểu phụ

+ Link MasterFields: Ghi các trường liên kết của báo biểu chính

+ Link Child Fields: Ghi các trường liên kết của báo biểu phụ

2.3. MACRO – CÁC HÀNH ĐỘNG CỦA MACRO TRONG CSDL

2.3.1. Khái niệm Macro

Macro là một đoạn chương trình, gồm 1 dãy các hành động, dùng để tự động hoá các thao tác và tổ chức giao diện chương trình.

Cấu trúc của 1 Macro bao gồm:

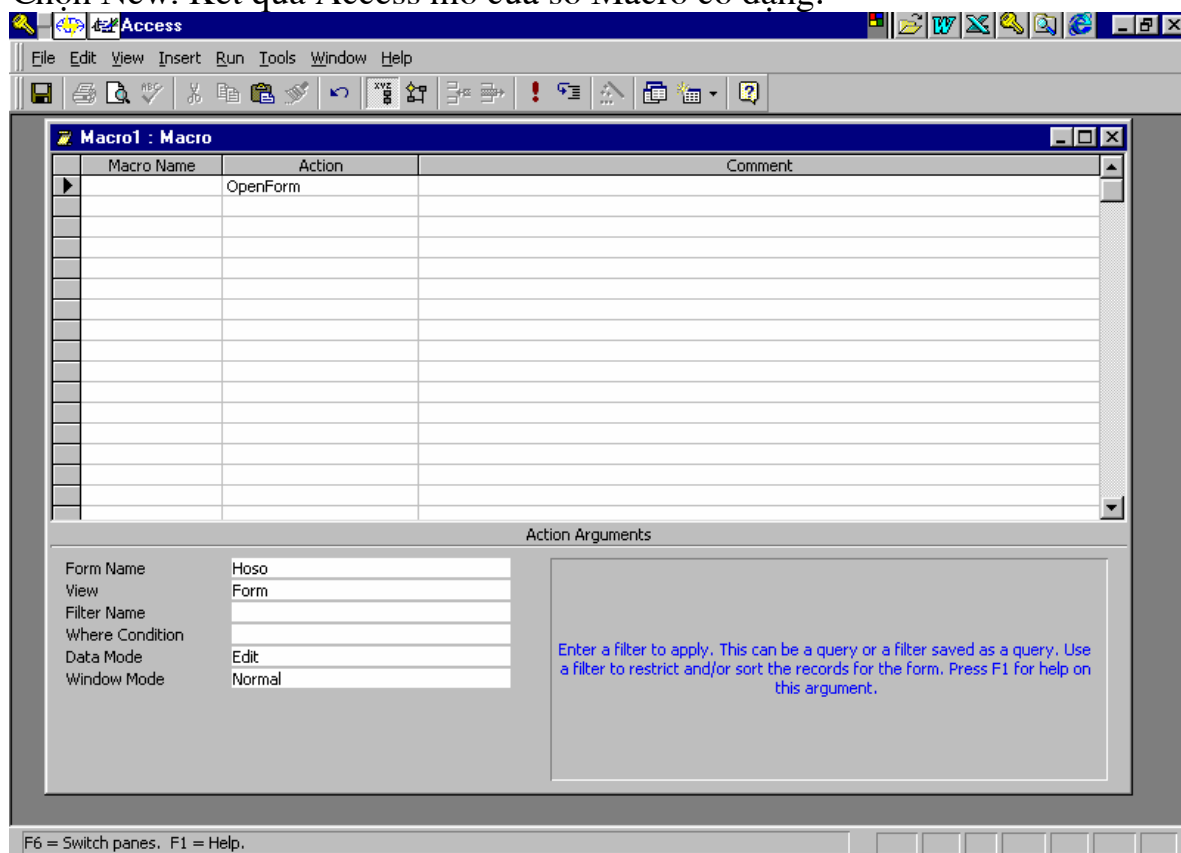
- 1/ Tên Action
- 2/ Các tham số

2.3.2. Tạo một Macro

Trình tự thao tác tạo 1 Macro tuân theo các bước sau:

B1: Trong cửa sổ Database chọn mục Macro

B2: Chọn New. Kết quả Access mở cửa sổ Macro có dạng:



Macro Name: cho phép đặt tên macro

Action: Các hành động của Macro

Comment: Mô tả các hành động của Macro

Thông thường Cửa sổ Macro không có cột Macro Name. Muốn có cột này vào menu View chọn Macro Name

B3: Chọn các hành động ở cột Action bằng cách:

+ Kích chuột tại mũi tên xuống bên phải ô Action để hiện danh sách các hành động

+ Chọn 1 hành động trong danh sách

B4: Xác định các đối tượng ứng cho hành động vừa chọn bằng cách:

+ Chuyển xuống phần dưới (dùng chuột hoặc bấm phím F6). Khi đó sẽ xuất hiện 1 số dòng ở phần dưới, mỗi dòng ứng với 1 đối

+ Đặt các đối tượng ứng vào các dòng theo 2 cách:

Cách 1: Gõ trực tiếp từ bàn phím

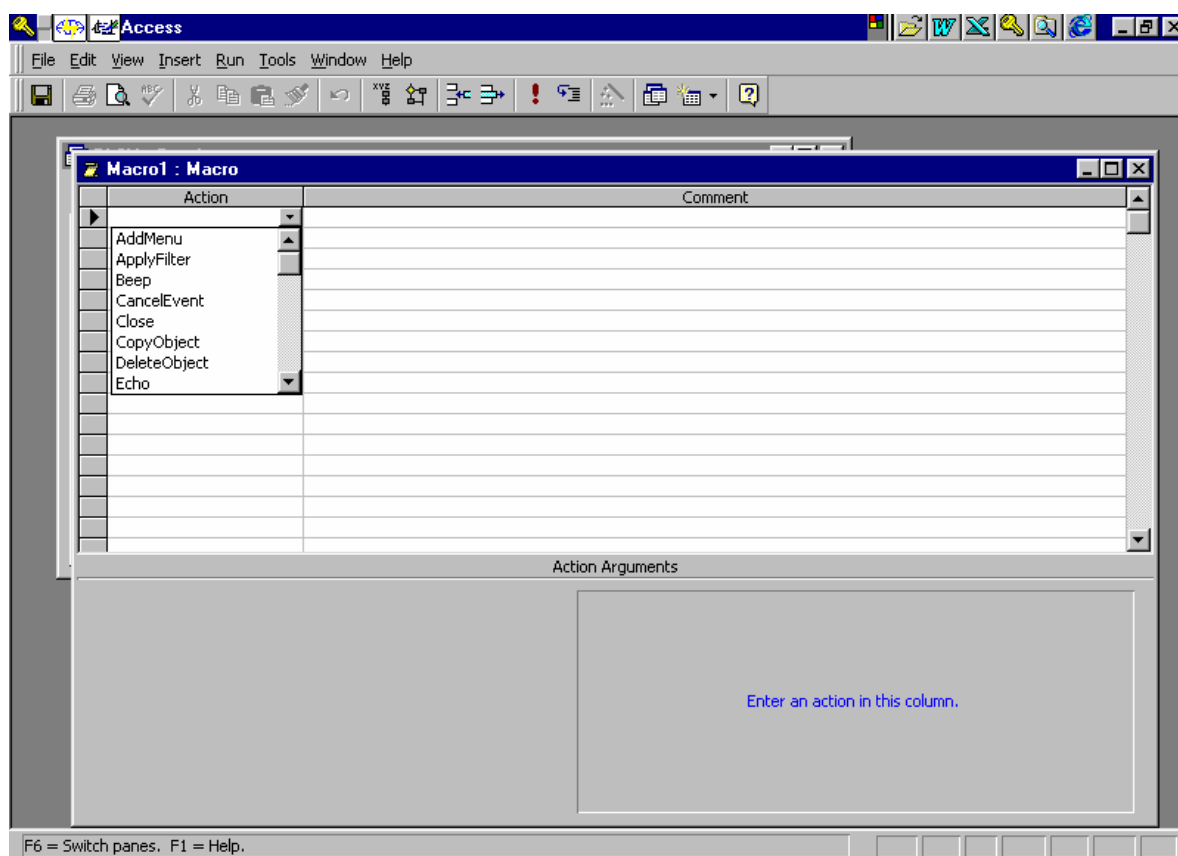
Cách 2: Kích chuột tại mũi tên xuống bên phải ô đang xét để mở danh sách giá trị, sau đó chọn giá trị cho đối.

2.3.3. Đưa các hành động vào Macro

Có hai phương pháp để đưa các hành động vào Macro:

Phương pháp 1:

1/ Bấm vào bên phải ô Action để hiện bảng danh sách hành động. Sau đó chọn các hành động trong danh sách



2/ Chuyển xuống nửa dưới để xác định các tham số của hành động

Phương pháp 2:

1/ Kéo một đối tượng (bảng, truy vấn, biểu mẫu, báo cáo, macro) từ cửa sổ Database vào một ô trong cột Action của cửa sổ Macro.

2/ Chuyển xuống nửa dưới để điều chỉnh các đối nếu cần

2.4.4. Sắp xếp lại các hành động

1. Di chuyển các hành động đến vị trí mới

+ Đánh dấu các hành động cần di chuyển

+ Dùng chuột kéo nó đến một vị trí mới

2. Cách xoá các hành động

+ Đánh dấu các hành động cần xoá

+ Bấm phím Delete

3. Cách ghi các Macro

Chọn Save từ menu File.

2.4.5. Thực hiện Macro

Có nhiều cách để thực hiện 1 macro

1. Khi đang thiết kế Macro

1/ Chọn mục Run từ menu Macro

2/ Bấm chuột tại biểu tượng Run

2. Tại cửa sổ Database:

1/ Chọn mục Macro để mở cửa sổ Macros

2/ Chọn Macro cần thực hiện

3/ Bấm nút Run

3. Từ các cửa sổ khác trong CSDL

1/ Chọn mục Run Macro từ menu File để hiện ra cửa sổ Run Macro

2/ Chọn Macro cần thực hiện rồi bấm OK

2.4.6. Thực hiện Macro từ một nút lệnh trên mẫu biểu

1. Cách tạo 1 nút lệnh gắn với một Macro

Cách 1:

+ Bấm Phím F11 để hiện cửa sổ Database

+ Kéo 1 Macro từ cửa sổ Database vào mẫu biểu đang thiết kế. Kết quả Access tạo ra 1 nút lệnh có:

Tiêu đề là tên Macro được kéo

Thuộc tính On Click: tên Macro

Cách 2:

1/ Chọn nút Command Button trên hộp công cụ

2/ Bấm chuột tại 1 vị trí trên mẫu biểu. Kết quả tạo ra 1 nút lệnh.

3/ Mở bảng thuộc tính của nút lệnh và:

+ Đặt lại tên tiêu đề của nút lệnh trong dòng thuộc tính Caption.

+ Trong dòng thuộc tính On Click: đưa vào tên Macro cần thực hiện

2.4.7. Sự kiện trong Access

1. Các loại sự kiện

1/ Các sự kiện xảy ra trên mẫu biểu, báo biểu (đóng, mở)

2/ Các sự kiện xảy ra trên các thành phần của mẫu/báo biểu

2. Cách nhận biết các sự kiện

Access sử dụng thuộc tính của các đối tượng để nhận biết và xử lý sự kiện xảy ra trên đối tượng. Mỗi sự kiện ứng với một thuộc tính.

3. Xử lý các sự kiện

+ Có thể gắn với 1 macro hoặc 1 thủ tục xử lý sự kiện

+ Khi sự kiện xảy ra thì macro hoặc thủ tục tương ứng sẽ được kích hoạt

+ Để gắn 1 macro hoặc 1 thủ tục vào 1 sự kiện thì ta chỉ cần đưa tên macro hoặc tên thủ tục vào dòng thuộc tính ứng với sự kiện.

4. Nút lệnh

+ Là cách thức hay dùng để kích hoạt 1 macro hoặc 1 thủ tục

+ Mỗi nút lệnh có 1 tập các sự kiện tác động lên nó.

+ Có thể gán mỗi sự kiện của nút lệnh với 1 macro hoặc 1 thủ tục

2.4.8. Thực hiện Macro bằng câu lệnh DOCMD

1. Cách viết câu lệnh Docmd

DoCmd Tên_hành_động Danh_sách_đối

+ Tên hành động: là tên một hành động của Macro

+ Danh sách đối: là 1 dãy các đối của Macro. Giữa 2 đối trong danh sách cách nhau bởi dấu phẩy

2. Phân loại đối của Macro

Có thể chia các đối của macro ra làm 2 loại:

1/ Đối do người dùng tự đặt như:

+ Tên các đối tượng (tên bảng, truy vấn, tên mẫu biểu, . . .)

+ Một điều kiện hay 1 biểu thức

2/ Đối nhận giá trị từ 1 danh sách quy định của Access

3. Cách viết các đối trong câu lệnh DoCmd

1/ Đối do người dùng tự đặt: Câu lệnh DoCmd biểu thị đối này bằng 1 biểu thức chuỗi.

2/ Đối nhận giá trị từ danh sách quy định của Access: thể hiện bằng số thứ tự của giá trị trong danh sách (đánh số từ 0)

3/ Khi bỏ qua 1 đối nhưng lại viết các đối sau, thì cần phải viết thêm các dấu phẩy để xác định số thứ tự của đối.

Chú ý:

+ Có những đối bắt buộc phải có như tên bảng, tên truy vấn, . . .

+ Một số đối có thể bỏ qua. Khi đó Access sẽ xử lý như sau:

+ Nếu đó là đối nhận giá trị từ danh sách quy định thì xem như đối có giá trị mặc định

+ Nếu đó là đối do người dùng tự đặt thì xem như không dùng đối này.

2.4.9. Một số hành động Macro hay dùng

1/ GotoControl

+ Công dụng: Dùng để chuyển con trỏ chuột và con trỏ bàn phím (chuyển Focus) đến 1 ô điều khiển của mẫu biểu hiện hành. Ô điều khiển được nhận Focus sẽ trở thành ô điều khiển hiện hành

+ Đối Control name: Tên ô điều khiển nhận Focus

2/ SetValue

+ Công dụng: Dùng để gán giá trị cho 1 điều khiển của mẫu biểu hiện hành

+ Các đối: có 2 đối:

Item: Tên của ô điều khiển nhận giá trị

Expression: Biểu thức dùng để gán

3/ FindRecord

+ Công dụng: Dùng để tìm 1 bản ghi trên mẫu biểu hiện hành. Bản ghi tìm được sẽ trở thành bản ghi hiện hành và hiển thị trên mẫu biểu

+ Các đối: Có 7 đối, nhưng thường dùng đối thứ nhất là:

Find What: Ghi 1 giá trị

+ Cách làm việc: Access sẽ tìm bản ghi đầu tiên mà trường gán với ô điều khiển hiện hành có giá trị ghi trong dòng đối FindWhat.

4/ **GotoRecord**

+ Công dụng: Di chuyển đến 1 bản ghi nào đó. Bản ghi được chuyển đến sẽ là bản ghi hiện hành.

+ Các đối: có 4 đối:

Object Type: Table, Query, Form

Object Name: tên đối tượng

Record: chọn trong danh sách (Previous, Next, Last, Goto, New)

Offset: N (giá trị nguyên)

5/ **DoMenuItem**

* Công dụng: Dùng để thực hiện 1 chức năng của 1 menu trong Access

* Các đối: có 4 đối

+ Menu Bar: Là 1 thanh menu của Access chứa chức năng cần thực hiện, được chọn trong danh sách: Form, Database, Filter, . . .

+ Menu Name: Là tên của menu chứa chức năng cần thực hiện

+ Command: Là tên của chức năng cần thực hiện

+ Sub Command: Là tên của chức năng con cần thực hiện

6/ **SendKeys**

+ Công dụng: Gửi 1 dãy các phím cho Access. Có thể dùng các hành động này thay thế cho 1 dãy các thao tác bấm phím.

+ Các đối: Có 2 đối là:

KeyStrokes: Là 1 dãy các phím cần cung cấp cho Access

Wait: Chọn No và Yes

Nếu chọn No thì thực hiện ngay Macro không cần đợi, nếu chọn Yes thì dừng macro chờ các phím được nhấn.

7/ **Quit**

+ Công dụng: Ra khỏi Access trở về Window

+ Đối Option: có thể nhận 1 trong 3 giá trị

Prompt: Nhắc người dùng ghi các thay đổi

Save all: Tự động ghi các thay đổi

Exit: không ghi các thay đổi.

8/ **Close**

+ Công dụng: Đóng 1 đối tượng đang mở

+ Object Type: chọn trong danh sách (Table, Query, Form, Report, Macro, Module)

+ Object name: tên đối tượng

9/ **Run Code**

+ Công dụng: gọi thực hiện 1 hàm, thủ tục

+ Đối Function Name: lời gọi đến hàm, thủ tục

10/ **Requery**

+ Công dụng: Để yêu cầu hiển thị nội dung mới nhất của ô điều khiển trên mẫu biểu hiện hành

+ Đối Control name: Tên ô điều khiển cần hiển thị

CHƯƠNG 3: XÂY DỰNG MODUL CHƯƠNG TRÌNH BẰNG ACCESS BASIC

3.1. CÁC KHÁI NIỆM CƠ BẢN VỀ ACCESS BASIC

NHỮNG QUI TẮC SOẠN THẢO VÀ THỰC HIỆN CHƯƠNG TRÌNH

3.1.1. Khái niệm cơ bản về Access Basic

Ngôn ngữ Access Basic là 1 ngôn ngữ lập trình hướng đối tượng theo các sự kiện trên những đối tượng giúp người sử dụng lập các module chương trình quản lý điều hành các sự kiện thông qua các thủ tục và hàm.

1. Tổ chức và sự hoạt động của chương trình

1/ Đặc điểm và vai trò

Các chương trình Access Basic được tổ chức và hoạt động trong môi trường Access nhằm xây dựng các ứng dụng Access phức tạp, hoàn chỉnh hơn, sau khi đã sử dụng các công cụ của Access như: Table, Query, Report, Form, Macro.

2/ Tổ chức chương trình

Chương trình bao gồm nhiều đơn thể (module) được phân loại như sau:

- + Đơn thể của mỗi mẫu biểu
- + Đơn thể của mỗi báo biểu
- + Đơn thể chung thuộc đối tượng module

Trên mỗi đơn thể có định nghĩa các biến, mảng, thủ tục và hàm.

3/ Hoạt động của chương trình

- + Không theo kiểu tuần tự như lập trình cấu trúc hệ thống
- + Hoạt động theo các sự kiện

4/ Phân loại các sự kiện

- + Sự kiện trên Form
- + Sự kiện trên Report
- + Sự kiện trên mỗi ô điều khiển của Form, Report

5/ Thủ tục xử lý sự kiện

- + Nằm trong các đơn thể của Form và Report
- + Gắn với 1 sự kiện nào đó và được kích hoạt
- + Khi thủ tục xử lý sự kiện được thực hiện nó có thể gọi tới các hàm, thủ tục

2 Đơn thể

1/ Đơn thể sử dụng chung có cấu trúc như sau:

Các tùy chọn

- + Option Base 1
- + Option explicit
- + Option Compare text binary

Khai báo các hằng, biến, mảng cấp đơn thể sử dụng từ khóa Dim

Các thủ tục phạm vi sử dụng cấp chương trình (Sub)

Các thủ tục của riêng đơn thể (Private Sub)

2/ Đơn thể của riêng mẫu/ báo biểu (Form/Report) có cấu trúc như sau:

Các tùy chọn

+ Option Base 1

+ Option explicit

+ Option Compare text binary

Khai báo các hằng, biến, mảng, . . . , cấp đơn thể

Các thủ tục xử lý sự kiện

Thủ tục tổng quát được gọi bởi thủ tục xử lý sự kiện tổng quát khác trong cùng đơn thể.

3/ Phạm vi sử dụng

Hằng, biến, mảng có 3 cấp sử dụng đó là:

+ Chương trình

+ Module

+ Thủ tục

Phạm vi sử dụng của các thủ tục trong đơn thể là toàn chương trình

Phạm vi sử dụng của các thủ tục trong mẫu/báo biểu chỉ trong phạm vi mẫu/báo biểu đó.

3. Cấu trúc của một hàm, thủ tục

Có 2 loại chương trình con:

+ Thủ tục hàm gọi là hàm

+ Thủ tục chương trình con

1- Thủ tục

Được khai báo như sau:

Sub Tên_thủ_tục (khai báo các đối)

End Sub

Chú ý: Không dùng cấu trúc khối như trong Pascal và C

Không có thủ tục trogn thủ tục

Lời gọi thủ tục: Tên_thủ_tục danh sách tham số

2- Hàm

Khai báo:

Function Tên_hàm(khai báo các đối) [as kiểu]

...

Tên_hàm= Giá trị

...

End Function

3- Thủ tục xử lý sự kiện

1/Sự kiện của Form

Sub Form Tên_sự_kiện()

...

End Sub

2/Sự kiện của Report

Sub Report Tên_sự_kiện ()

...

End Sub

3/ Sự kiện của ô điều khiển

Sub Tên_ô_điều_khiển Tên_sự_kiến ()

...

End Sub

Chú ý: Tên của thủ tục xử lý sự kiện là do hệ thống tự xác định.

4- Dịch, kiểm tra và chạy thử chương trình

1/ Các thủ tục, hàm có thể biên soạn trong các đơn thể của mẫu biểu, báo biểu hoặc trong các đơn thể sử dụng chung

2/ Trong khi soạn thảo, nếu chuyển sang dòng khác thì Access sẽ kiểm tra ngữ pháp của dòng lệnh vừa soạn và đưa ra các dòng thông báo sai nếu có.

3/ Để dịch kiểm tra lỗi của một hàm, thủ tục ta sử dụng chức năng “Compile loaded modules” của menu Run.

3.1.2. Vào ra và các quy tắc cần biết khi soạn thảo, thực hiện chương trình

1. Các hàm **INPUTBOX** và **INPUTBOX\$**

1/ Hàm InputBox trả về một giá trị kiểu chuỗi

2/ Hộp hội thoại nhập dữ liệu trong hàm Input Box gồm các thành phần sau:

+ Tiêu đề (Title)

+ Lời nhắc (Prompt)

+ Hộp văn bản

+ Các nút OK và Cancel

3/ Cú pháp các hàm InputBox và InputBox\$:

InputBox [\$] (Lời nhắc [, [Tiêu_đề] [, [Giá trị mặc định] [, X_hộp, Y_hộp]])

4/ Ý nghĩa các đối

+ Lời nhắc: Là 1 biểu thức chuỗi dùng để nhắc nhở, hướng dẫn nhập liệu (không quá 255 ký tự)

+ Tiêu đề : Biểu thức chuỗi hiện trên thanh tiêu đề của hộp hội thoại

+ Giá trị mặc định: là biểu thức chuỗi hiện trên hộp văn bản

+ X_hộp, Y_hộp: là tọa độ tính bằng Twips của điểm góc trên/trái của hộp hội thoại

5/ Các đối tùy chọn

Trong các đối trên chỉ đối thứ nhất là bắt buộc phải có, các đối còn lại có thể bỏ qua.

+ Nếu bỏ qua tiêu đề thì thanh tiêu đề sẽ bị trống

+ Nếu bỏ qua giá trị mặc định thì hộp văn bản sẽ để trống

+ Nếu bỏ qua X_hộp, Y_hộp thì hộp hội thoại sẽ nằm chính giữa theo chiều ngang và một phần ba phía trên theo chiều dọc

6/ Tác dụng của các nút OK, Cancel

+ Khi người dùng chọn nút OK thì hàm trả về giá trị có trong hộp văn bản

+ Khi người dùng chọn nút Cancel thì hàm trả về chuỗi có độ dài không

2. Hàm **MSGBOX** và thủ tục **MSGBOX**

1/ Giới thiệu

Thủ tục MsgBox dùng để đưa một biểu thức chuỗi ra hộp hội thoại

Hàm MsgBox dùng để đưa một biểu thức chuỗi ra hộp hội thoại và nhận một giá trị số trả về để biết người dùng chọn nút nào trong các nút OK, Cancel, Retry, . . .)

2/ Cú pháp của hàm, thủ tục MsgBox

Hàm:

MsgBox (nd [, kiểu [, tiêu_đề]])

Thủ tục:

MsgBox nd[, kiểu [, tiêu_đề]]

3/ Ý nghĩa các đối

+ nd: Là 1 biểu thức chuỗi chứa nội dung trong hộp hội thoại

+ tiêu_đề: Là một biểu thức chuỗi được hiện trong thanh tiêu đề của hộp hội thoại. Nếu bỏ qua đối này thì tiêu đề trên hộp sẽ mặc định là Microsoft Access

+ Kiểu: là 1 tổng các số nguyên để xác định:

. Các nút sẽ hiển thị

. Các biểu tượng

. Nút mặc định sẽ được chọn

Nhóm 1: Các nút

Giá trị	Ý nghĩa
0	Hiện chỉ nút OK
1	Hiện các nút OK và Cancel
2	Hiện các nút Abort, Retry và Ignore
3	Hiện các nút Yes và No và Cancel
4	Hiện các nút Yes và No
5	Hiện các nút Retry và Cancel

Nhóm 2: Các biểu tượng

Giá trị	Ý nghĩa
0	Không hiện biểu tượng
16	Hiện biểu tượng Critical Message
32	Hiện biểu tượng Warning Query
48	Hiện biểu tượng Warning Message
64	Hiện biểu tượng Information Message

Nhóm 3: Nút mặc định được chọn

Giá trị	Ý nghĩa
0	Nút đầu tiên
256	Nút thứ hai
512	Nút thứ ba

Ví dụ: Kiểu 4 + 16 + 256 thì hộp hội thoại sẽ hiện 2 nút Yes và No, biểu tượng Critical Message và nút No được chọn

4/ Giá trị của hàm

Giá trị	Nút được chọn
1	OK

2	Cancel
3	Abort
4	Retry
5	Ignore
6	Yes
7	No

3. Một số quy tắc khi viết chương trình

1/ Câu lệnh

+ Một câu lệnh chỉ viết trên 1 dòng. Câu lệnh không thể tiếp tục từ dòng này sang dòng khác

+ Để viết nhiều câu lệnh trên 1 dòng ta dùng dấu hai chấm để ngăn cách 2 câu lệnh

2/ Ghi chú

Để viết lời giải thích, ghi chú ta có thể dùng cách sau:

+ Mở đầu bằng dấu từ khoá Rem hoặc dấu nháy đơn

+ Phần còn lại của 1 dòng viết sau dấu nháy đơn được xem là ghi chú.

4. Dịch và chạy chương trình trong cửa sổ trực diện

1/ Soạn thảo thủ tục

Soạn thảo 1 thủ tục trong cửa sổ của 1 đơn thể chung hay mẫu/báo biểu

2/ Thực hiện thủ tục: Có 2 cách:

Cách 1: Gắn thủ tục với sự kiện của 1 nút lệnh nào đó trong 1 mẫu/báo biểu theo hành động.

Cách 2: Soạn thảo thủ tục trong 1 đơn thể bất kỳ. Sau đó mở cửa sổ trực diện Immediate Window thực hiện thủ tục trong cửa sổ này

3/ Mở cửa sổ Immediate Window

Vào menu View, chọn Immediate Window

4/ Dịch, kiểm tra và chạy thử chương trình

Sau khi biên soạn 1 thủ tục ta có thể dịch và kiểm tra lỗi của nó bằng cách dùng menu Run, chức năng Compile hoặc kích chuột tại biểu tượng Compile.

5. Đối dữ liệu số sang chuỗi

Dùng 2 hàm Str và Format

1/ Hàm Str

Cú pháp: Str(biểu thức số)

VD: MsgBox "Thuong=" & str(100/3)

2/ Hàm Format

Đổi số sang chuỗi và định dạng chuỗi để in ra theo ý muốn của người dùng

Cú pháp: Format (Biểu thức số, chuỗi định dạng)

VD: MsgBox "Thuong=" & Format(100/3, "0.00")

3. 2. CÁC TOÁN TỬ ĐIỀU KHIỂN – ĐỐI TƯỢNG VÀ BIẾN ĐỐI TƯỢNG

3.2.1. Quy tắc đặt tên biến

Tên biến (mảng, hàm, . . .) cần được đặt theo các qui định sau:

- + Bao gồm các chữ cái, chữ số và dấu gạch nối
- + Kí tự đầu phải là chữ cái
- + Không được trùng với các từ khoá của Access
- + Độ dài không quá 40 ký tự

3.2.2. Khai báo ngầm định các biến

Nếu trong phần Declaration Section ta viết dòng lệnh:

Option Explicit

thì không cho phép khai báo các biến ngầm định nghĩa là phải khai báo các biến tường minh thông qua câu lệnh Dim, Global

Nếu không có câu lệnh đó thì được phép khai báo các biến ngầm định

1/ Cách khai báo các biến ngầm định

Biến khai báo ngầm định là biến mới có kiểu Variant có thể nhận các giá trị:

- + Kiểu số (Number)
- + Kiểu chuỗi (String)
- + Kiểu Date/time
- + Giá trị Null (rỗng)

2/ Định tên biến cho các biến ngầm định

Có 2 cách định tên biến cho các biến ngầm định:

Cách 1:

Dùng các ký tự định kiểu viết sau tên biến, đó là các ký tự

% Integer & Long ! Single # Double @ Currency \$ String

Ví dụ:

SI % là 1 biến kiểu Integer; SL & là biến kiểu Long , . . .

Cách 2:

Dùng các toán tử định dạng

Def Kiểu Chữ cái

Def Kiểu Chữ cái - Chữ cái

để quy định kiểu của biến theo chữ đầu của biến

VD: DefStr S quy định các biến bắt đầu bằng chữ S có kiểu String

DefInt I-M quy định các biến mà chữ đầu I-M có kiểu Integer

Kiểu trong toán tử DefKiểu gồm:

Int (Integer), Lng (Long), Sng(Single), Dbl(Double), Cur(Currency),
Str(String), Var (Variant)

Chú ý: Toán tử DefKiểu cần viết trong phần “Declaration Section” của đơn thể và nó có tác dụng trong tất cả các hàm và thủ tục của đơn thể

3.2.3. Khai báo tường minh

Các hàm kiểm tra giá trị biến Variant bao gồm:

Isnumeric(x), Isdate(x), Isnull(x), Isempty(x)

2/ Các kiểu dữ liệu cơ bản khác

<i>Tên</i>	<i>Mô tả</i>
Integer	2 byte
Long	4 byte
Single	4 byte, dấu chấm động với 7 chữ số có nghĩa
Double	8 byte, dấu chấm động với 15 chữ số có nghĩa
Currency	8 byte, dấu chấm cố định với 15 chữ số phần nguyên, 4 chữ số phần thập phân
String	Chiều dài 0 đến 65500 kí tự
Variant	Số dấu chấm động ngày/giờ, chuỗi hoặc null

Chú ý:

+ Chuỗi có độ dài biến đổi: khai báo: biến as string

+ Chuỗi có độ dài cố định : khai báo: Biến as string * độ dài (là 1 số nguyên)

3.2.8. Hằng ký hiệu

1/ Khai báo:

[Global] Const Tên_hằng=biểu_thức, tên_hằng=biểu_thức, . . .

2/ Phạm vi sử dụng

Giống như các biến:

+ Cấp thủ tục: khai báo trong thủ tục

+ Cấp đơn thể: khai báo trong phần declaration section của đơn thể

+ Cấp chương trình : khai báo Global const trong phần Declaration section của đơn thể

3.2.9. Mảng

1/Cách khai báo

{	Global	Tên_mảng (cs1, cs2, . . .) [as type]
	Dim	
	Static	
	Redim	

Có 2 cách xác định miền giá trị chỉ số:

Cách 1:

Dùng Option base xác định cận dưới, trong khai báo chỉ ra cận trên

VD: Option base 1

Dim a(5) ' chỉ số từ 1 đến 5

Cách 2:

Dùng cách viết: Cận_dưới to Cận_trên

VD: Dim a(-2 to 2, 1 to 6)

Mảng 2 chiều: chỉ số 1 chạy từ -2 đến 2, chỉ số 2 chạy từ 1 đến 6.

Chú ý: Các cận phải là hằng số nguyên và có thể nhận giá trị âm trong khoảng -32768 đến 32767

Số chiều của mảng có thể lên đến 60

2/ Phạm vi sử dụng

Giống như biến:

+ Cấp chương trình: khai báo trong phần Declaration section của đơn thể bằng từ khoá Global

VD: Global A(5, 3 to 7)

+ Cấp đơn thể: Khai báo trong phần Declaration section của đơn thể bằng từ khoá Dim

VD: Dim B(7, 8)

+ Cấp thủ tục: Khai báo trong thủ tục bằng từ khoá Static và Redim

VD: Static X(6)

Redim Y(7)

3/ Xác định kiểu mảng

Các cách xác định kiểu biến đều có thể dùng để xác định kiểu mảng:

+ Dùng kí tự định dạng viết sau tên mảng

+ Dùng toán tử Defkiểu

+ Dùng mệnh đề as kiểu trong khai báo mảng

3.2.10. Kiểu tự tạo

Kiểu tự tạo tương tự như kiểu bản ghi của Turbo Pascal bao gồm nhiều thành phần, được định nghĩa thông qua từ khoá Type trong phần Declaration section:

Type Tên_kiểu

Tên_thành_phần as Kiểu

[Tên_thành_phần as Kiểu]

.....

End Type

Trong đó:

+Tên_kiểu và Tên_thành_phần được đặt theo quy tắc của tên biến

+ Kiểu có thể bao gồm các kiểu dữ liệu cơ bản và các kiểu dữ liệu đã định nghĩa

VD:

TYPE Hocsinh

Hoten as String*25

NS as Variant

Toan as single: Ly as Single: Hoa as Single

END TYPE

2/ Khai báo các biến mảng kiểu tự tạo

Tương tự như khai báo đối với các biến, mảng kiểu chuẩn

VD: Dim hs as Hocsinh

Static DS(1 to 100) as Hocsinh

3/ Truy nhập đến các thành phần của biến, mảng kiểu tự tạo

Cú pháp:

Tên_biến.tên_thành_phần

Tên_mảng(chỉ số).Tên_thành_phần

3.2.11. Các phép toán và biểu thức

1/ Các phép toán số học

Bao gồm các phép toán sau đây:

^ : lũy thừa * : nhân / : chia \ : chia nguyên mod : lấy phần dư
+ : cộng - : trừ

2/ Các phép toán so sánh

Bao gồm các phép: >, >=, <, <=, =, <>

Các phép toán so sánh cho kết quả là -1 nếu đúng, là 0 nếu sai

3/ Phép ghép chuỗi

Cú pháp:

Chuỗi_1 & chuỗi_2 & ... & chuỗi_n

Chuỗi_1 + chuỗi_2 + ... + chuỗi_n

Sẽ cho một chuỗi mới bằng cách ghép các chuỗi thành phần lại

Phép & linh hoạt hơn phép +, nhờ cho phép tự động chuyển các toán hạng từ dạng số sang dạng chuỗi rồi mới tiến hành ghép, còn phép + chỉ thực hiện trên các toán hạng dạng chuỗi.

4/ Các phép toán logic

Bao gồm: And, Or, Xor, Eqv, Imp

Cho phép thực hiện trên các giá trị logic (-1 và 0) hoặc trên các số nguyên theo quy tắc trong bảng:

x	y	X and Y	X or Y	X Xor Y	X Eqv Y	X Imp Y
0	0	0	0	0	-1	-1
0	-1	0	-1	-1	0	-1
-1	0	0	-1	-1	0	0
-1	-1	-1	-1	0	-1	-1

5/ Các hàm thường dùng

5.1/ Các hàm cho dữ liệu dạng số

1- Các hàm chuyển đổi số nguyên

Fix(x) cho giá trị nguyên là phần nguyên của x

VD: Fix(3.5) = 3

2- Int(x) cho giá trị nguyên lớn nhất không vượt quá x

Vd: int(-6.7) = -7

2- Các hàm toán học

+ Exp(x) : cho kết quả của E mũ x

+ Log(x) tính logarit cơ số e của x

+ Sqr(x) tính căn bậc hai của x

3- Các hàm lượng giác

+ Atn(x) tính arctan của x

+ Tan(x) tính tang của x

+ Sin(x) tính sin của x

+ Cos(x) tính cos của x

5.2/ Các hàm cho dữ liệu dạng chuỗi

1- Hàm Len

Cú pháp: Len(biểu_thức_chuỗi) cho biết độ dài chuỗi

2- Hàm Left

Cú pháp: Left[\$](biểu_thức_chuỗi, n) cho chuỗi con gồm n kí tự ở bên trái của biểu thức chuỗi.

3- Hàm Right

Cú pháp: Right[\$](biểu_thức_chuỗi, n) cho chuỗi con gồm n kí tự bên phải của biểu_thức_chuỗi

4- Hàm Mid

Cú pháp: Mid [\$](biểu_thức_chuỗi, m, n) trích từ biểu thức chuỗi một chuỗi con có độ dài là n bắt đầu từ vị trí thứ m.

3/ Các hàm dùng cho dữ liệu dạng ngày giờ

+ Date() cho biết ngày hiện tại của hệ thống

+ Date\$() cho kết quả kiểu chuỗi của ngày hệ thống

+ Time() cho kết quả giờ hệ thống kiểu Variant

+ Time\$() cho kết quả chuỗi chỉ giờ hệ thống

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

4/ Hàm lựa chọn IIF

Cú pháp : IIF(điều_kiện, Giá_trị_đúng, Giá_trị_sai)

VD: MsgBox IIF(10 mod 2 =0, “Số chẵn”, “số lẻ”) sẽ cho thông báo “Số chẵn”

5/ Hàm tìm kiếm theo mẫu

Cú pháp: Biểu_thức Like Mẫu

Sẽ cho kết quả đúng nếu biểu thức chuỗi thuộc mẫu và ngược lại

Mẫu là 1 biểu thức chuỗi bao gồm các ký tự, cho phép dùng các ký tự thế chân như:

* đại diện cho 1 chuỗi kí tự bất kỳ

? đại diện cho 1 kí tự bất kỳ

đại diện cho 1 chữ số bất kỳ

[kí_tự_đầu – kí_tự_cuối] từ ký tự đầu đến kí tự cuối

[!kí_tự_đầu – kí_tự_cuối] không thuộc từ kí tự đầu đến kí tự cuối

VD: Hoten like “H*g” sẽ cho biết toàn bộ các xâu kí tự chứa hoten có kí tự đầu là H, kí tự cuối là g.

3.2.12. Các toán tử điều khiển

1. Toán tử rẽ nhánh IF ... THEN

Dạng 1: Hiệu lệnh trên một dòng

Cú pháp:

IF điều_kiện THEN hiệu_lệnh

Dạng 2: Hiệu lệnh trên nhiều dòng

Cú pháp:

IF điều_kiện THEN

Các_câu_lệnh

END IF

Dạng 3: Rẽ nhiều nhánh

Cú pháp:

If điều_kiện_1 Then

Nhóm lệnh 1

else

If điều_kiện_2 Then

Nhóm lệnh 2

Else

Nhóm lệnh 3

End if

End if

2. Toán tử lựa chọn SELECT CASE

1/ Cú pháp:

Select Case Biểu_thức_nguyên

Case danh_sách_giá_trị_1:

Nhóm_lệnh_1

Case danh_sách_giá_trị_2:

Nhóm lệnh 2

.....

Case danh_sách_giá_trị_n:

Nhóm lệnh n

[Case else

Nhóm lệnh n+1]

End Select

2/ Cách viết danh sách các giá trị

Được ngăn cách với nhau bởi dấu phẩy, không phụ thuộc vào thứ tự. Đối với giá trị liên tiếp từ n1 đến n2 có thể viết n1 to n2

3/ Nguyên tắc hoạt động

Nếu danh_sách_giá_trị_i là danh sách đầu tiên chứa biểu_thức_nguyên thì chương trình thực hiện nhóm lệnh i

Nếu không có danh_sách_giá_trị nào chứa biểu_thức_nguyên thì :

+ Chương trình thực hiện nhóm lệnh sau Case else (nếu có)

+ Hoặc ra khỏi Select Case (nếu không có Case else)

3. Toán tử For

1/ Cú pháp:

For Biến_đếm = giá_trị_đầu to giá_trị_cuối [step bước]
[Exit For]

...

Next

2/ Ghi chú

+ Bước có thể dương hoặc âm

+ Nếu bỏ Step bước thì bước bằng 1

+ Toán tử Exit For dùng để thoát khỏi chu trình

3/ Hoạt động khi bước dương (> 0)

+ biến_đếm = giá_trị_đầu

+ Nếu biến_đếm <= giá_trị_cuối thì

Tiếp tục chu trình

Biến_đếm = biến_đếm + bước

ngược lại

thoát khỏi vòng lặp

4/ Trường hợp bước âm ngược lại

4. Toán tử Do ... Loop với điều kiện trước

Dạng 1:

Cú pháp:

Do While điều_kiện_tiếp_tục_chu_trình

...

[exit do]

...

Loop

Dạng 2:

Cú pháp:

Do Until điều_kiện_ra_khỏi_chu_trình

...

[exit do]

...

Loop

5. Toán tử Do ... Loop với điều kiện sau

Dạng 1:

Cú pháp:

Do

...

[exit do]

...

Loop While điều_kiện_tiếp_tục_chu_trình

Dạng 2:

Cú pháp:

Do

...

[exit do]

...

Loop Until điều_kiện_kết_thúc_chu_trình

6. Các lệnh nhảy và kết thúc chương trình

1/ Nhảy không điều kiện

Cú pháp: Goto nhãn

2/ Nhảy tới đoạn chương trình con

Cú pháp: Gosub nhãn

...

nhãn;

...

Return

3/ Nhảy có lựa chọn

Cú pháp: On bt Goto/gosub nhãn_1, ..., nhãn_n

trong đó bt là 1 biểu thức nguyên thoả mãn $1 \leq bt \leq 255$

Bt=i thì nhảy tới nhãn i

4/ Nhảy khi có lỗi

Cú pháp:

On Error goto nhãn

...

exit sub

nhãn:

Xử lý lỗi

Resume Next
nhấn vào đó

5/ Kết thúc chương trình

Cú pháp: End

3.2.13. Thực hiện Macro trong chương trình

1. Câu lệnh Docmd

1/Cú pháp:

DocCmd Tên_hành_động [danh sách đối]

2/ Có 2 loại đối:

+ Tên của 1 đối tượng do người dùng tự đặt: tên bảng, tên mẫu biểu, . . . phải nhập từ bàn phím

+ Đối tượng lựa chọn từ 1 danh sách có sẵn

3/ Cách ghi các đối trong câu lệnh DoCmd

Đối do người dùng tự đặt được biểu thị bằng 1 biểu thức chuỗi chứa tên đối tượng

Đối chọn trong danh sách chuẩn được biểu thị bằng 1 số nguyên là số thứ tự của dòng trong danh sách

4/ Có thể bỏ qua các đối trong câu lệnh DoCmd

2. Đóng các đối tượng

Cú pháp:

Docmd.Close [Object Type], [ObjectName]

+Object Type: có thể chọn trong danh sách:

0 Table

1 Query

2 Form

3 Report

4 Macro

5 Module

+ Object Name: là tên đối tượng do người dùng tự đặt

3. Mở các đối tượng

1/ Hành động OpenForm

Cú pháp:

Docmd.OpenForm (FormName, [View], [FilterName], [Where Condition], [Data mode], [window mode])

+ FormName: tên biểu thức chuỗi chứa tên mẫu biểu cần mở

+ View: chọn trong danh sách;

0 Form

1 Design

2 PrintPreview

3 Datasheet

+ Filter Name: là biểu thức chuỗi chứa tên của 1 Filter dùng để sắp xếp hoặc lọc các thông tin cần thiết

+ Where Condition: biểu thức chứa chuỗi điều kiện chọn lọc thông tin

+ Data Mode: chọn trong danh sách:

- 0 Add
- 1 Edit
- 2 Read Only

+ Window Mode : chọn trong danh sách

- 0 Normal
- 1 Hidden
- 2 Icon
- 3 Dialog

2/ Hành động OpenTable

Dùng để mở 1 bảng dữ liệu.

Cú pháp:

Docmd.OpenTable TableName, [View], [DataMode]

+TableName: chứa tên bảng cần mở

+ View: chọn trong danh sách;

- 0 Datasheet
- 1 Design
- 2 Print Preview

+ Data Mode: chọn trong danh sách:

- 0 Add
- 1 Edit
- 2 Read Only

3/ Hành động OpenQuery

Dùng để mở 1 truy vấn

Cú pháp:

Docmd.OpenQuery QueryName, [View], [Data Mode]

+QueryName: chứa tên truy vấn cần mở

+ View: chọn trong danh sách;

- 3 Datasheet
- 4 Design
- 5 Print Preview

+ Data Mode: chọn trong danh sách:

- 0 Add
- 1 Edit
- 2 Read Only

4/Hành động OpenReport

Dùng để mở 1 báo biểu

Cú pháp:

Docmd.OpenReport (ReportName, [View], [FilterName], [Where Condition])

+ ReportName: tên biểu thức chuỗi chứa tên báo biểu cần mở

+ View: chọn trong danh sách;

- 0 Print
- 1 Design
- 2 Print Preview

+ Filter Name: là biểu thức chuỗi chứa tên của 1 Filter dùng để sắp xếp hoặc lọc các thông tin cần thiết

+ Where Condition: biểu thức chứa chuỗi điều kiện chọn lọc thông tin

5/ Hành động OpenModule

Cú pháp:

Docmd.OpenModule ModuleName, [ProcedureName]

+ ModuleName: là biểu thức chuỗi chứa tên Module cần mở

+ ProcedureName: là biểu thức chuỗi chứa thủ tục của đơn thể

4. Các hành động trên các ô điều khiển

1/ Sơn lại các ô điều khiển

Dùng để thể hiện nội dung mới nhất của 1 ô điều khiển mà nó vừa thay đổi

Cú pháp:

DoCmd.RepaintObject ObjectType, ObjectName

+ ObjectType:

0 Table

1 Query

2 Form

3 Report

4 Macro

5 Module

+ ObjectName: Là biểu thức chuỗi chứa tên đối tượng cần sơn lại

2/ Đặt con trỏ chuột tại 1 ô điều khiển

Đưa con trỏ chuột hoặc con trỏ bàn phím đến 1 ô điều khiển trong mẫu biểu hiện hành

Cú pháp:

DoCmd.GotoControl “Tên ô điều khiển”

3/ Hiện nội dung vừa thay đổi của 1 ô điều khiển

Hiện nội dung mới nhất của 1 ô điều khiển vừa được thay đổi

Cú pháp:

DoCmd.Requery “Tên ô điều khiển”

3.2.14. Một số đối tượng và tuyển tập đối tượng thường dùng

1. Kiểu đối tượng (Object Type)

Các kiểu đối tượng thường dùng trong MS Access bao gồm: DBEngine, Workspace, Database, Form, Report, TableDef, QueryDef, RecordSet, Control, Field, Index.

2. Đối tượng (Object)

Mỗi kiểu đối tượng có nhiều đối tượng cụ thể. VD mẫu biểu có kiểu Form, báo biểu có kiểu Report, . . .

3. Biến đối tượng (Object Variable)

Là biến biểu thị đối tượng, nó được khai báo bằng cách dùng tên các đối tượng nêu trên.

+ Cách khai báo:

Dim tên_biến as Kiểu_đối_tượng

VD: Dim DB as Database, Rec as Recordset, FM as Form, R as Report. . .

+ Tham chiếu đến 1 biến đối tượng cùng kiểu:

Set tên_biến = dạng_biểu_thị_đối_tượng

VD: Set FM = Forms![Hoso]

+ Sau khi đã gắn 1 biến với 1 đối tượng có thể dùng biến để biểu thị đối tượng này

VD:

+ Biểu thị thuộc tính Visible của mẫu biểu Hoso sau khi đã gắn nó với biến FM :

Set FM = Forms![Hoso]

FM.Visible (thay cho câu lệnh Forms![Hoso].Visible)

+ Biểu thị 1 ô điều khiển trong mẫu biểu Hoso

FM![Hoten]

4. Tuyển tập đối tượng

Là tập hợp các đối tượng có cùng loại. Để biểu thị tuyển tập đối tượng ta chỉ cần thêm vào chữ “s” đằng sau tên kiểu đối tượng (trừ DBEngine).

Tuyển tập đối tượng chính là 1 đối tượng tiền định của Access đã được định nghĩa trước, nó có nhiều thành phần (các đối tượng con)

3.2.15. Biểu thị đối tượng và các thành phần của đối tượng

1. Các thành phần của 1 đối tượng

Chia làm 3 loại:

+ Đối tượng con (Sub Object)

+ Thuộc tính (Property)

+ Phương thức (Method)

VD: Đối tượng Forms có các đối tượng con là các mẫu biểu trong CSDL

2. Đối tượng khởi thủy

Là đối tượng không nằm trong bất kỳ đối tượng nào khác, có 3 đối tượng khởi thủy là:

+ Forms : có các đối tượng con là các mẫu biểu

+ Reports: có các đối tượng con là các báo biểu

+ DBEngine : các đối tượng thành phần là các vùng làm việc kiểu Workspace

3. Biểu thị thành phần đối tượng

Để biểu thị 1 thành phần của 1 đối tượng bao gồm : đối tượng, thuộc tính, phương thức ta luôn phải bắt đầu từ đối tượng khởi thủy

+ Biểu thị mẫu biểu: Forms![tên_mẫu_biểu] hoặc Forms(tên_mẫu_biểu)

VD: Forms![Hoso] hoặc Forms(“Hoso”)

+ Biểu thị thuộc tính: Forms![tên_mẫu_biểu].thuộc_tính

VD: Forms![Hoso].Visible

4. Quy tắc biểu thị thành phần của đối tượng

Để biểu thị thành phần của đối tượng bắt đầu từ đối tượng khởi thủy, sau đó là danh sách các đối tượng có quan hệ phụ thuộc, cuối cùng là thành phần cần biểu thị. Giữa 2 đối tượng trong danh sách được kết nối bằng:

- + Dấu chấm than “!” nếu đối tượng do người dùng định nghĩa
- + Dấu chấm trong các trường hợp: đối tượng là thuộc tính, phương thức hay đối tượng tiên định

5. Chú ý

+ Để liên kết 1 tuyển tập đối tượng với 1 đối tượng thành phần có thể dùng thêm 3 cách:

Tuyển tập![tên_đối_tượng]

Tuyển_tập(“tên_đối_tượng)

Tuyển_tập(số_thứ_tự)

+ Có thể thay dấu chấm than bằng dấu chấm để liên kết các đối tượng không tiên định

+ Tên các đối tượng không tiên định cho phép chứa các khoảng trống thì cần được bao bọc trong dấu ngoặc vuông ([tên])

3.2.15. Mẫu biểu - Mối quan hệ phân cấp trong mẫu biểu

1. Sơ đồ quan hệ

Mức 1: Đối tượng khởi thủy Forms

Mức 2: Các mẫu biểu đang mở là thành phần của Forms

Mức 3: Các ô điều khiển (Control) là thành phần của mẫu biểu

Mức 4: Đối tượng tiên định form là thành phần của ô điều khiển kiểu SubForm

Mức 5: Mẫu biểu con là thành phần của SubForm

2. Truy nhập đến các thành phần của mẫu biểu

+ Biểu diễn mẫu biểu

Forms![tên_mẫu_biểu]

VD: Forms![Hoso]

+ Biểu diễn thuộc tính

Forms![tên_mẫu_biểu].tên_thuộc_tính

VD: Forms![Hoso].Visible

+ Biểu diễn 1 ô điều khiển

Forms![tên_mẫu_biểu]![tên_ô_điều_khiển]

VD: Forms![Hoso].[Hoten]

+ Biểu thị mẫu biểu con gắn với 1 ô điều khiển

Forms![Tên_mẫu_biểu_chính]![tên_ô_điều_khiển].Form![mẫu_biểu_con]

VD: Forms![HosoHS]![Ht].Forms![Hoso]

3. Thuộc tính

Các thuộc tính thường sử dụng trong mẫu biểu:

+ Name: Cho biết tên đối tượng

+ Visible: Quy định hiển thị hay không hiển thị 1 mẫu biểu hay 1 ô điều khiển

+ Count: Xác định số đối tượng thành phần của tuyển tập

3.2.16. Báo biểu - Môi quan hệ phân cấp trong đối tượng Report

1. Sơ đồ quan hệ phân cấp

Mức 1: Đối tượng khởi thủy Reports

Mức 2: Các mẫu biểu đang mở là thành phần của Reports

Mức 3: Các ô điều khiển (Control) là thành phần của báo biểu

Mức 4: Đối tượng tiền định form là thành phần của ô điều khiển kiểu

SubReport

Mức 5: Báo biểu con là thành phần của Report

2. Truy nhập đến các thành phần của mẫu biểu

Tương tự như truy nhập đến các thành phần của Forms

3. Thuộc tính

+ Name: Cho biết tên đối tượng

+ Visible: Quy định hiển thị hay không hiển thị 1 báo biểu hay 1 ô điều khiển

+ Count: Xác định số đối tượng thành phần của tuyến tập