

MICROSOFT VISUAL FOXPRO 6.0

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU 1

Người thực hiện: Phan Đình Sinh

MICROSOFT VISUAL FOXPRO 6.0

- **Thời gian:** 60 tiết Lý thuyết và bài tập + 30 tiết Thực hành
- **Mục đích:**
 - Trang bị cho học viên cách sử dụng và khai thác Visual Foxpro.
 - Các khái niệm về CSDL, Hệ Quản trị CSDL và các thao tác trên CSDL.
 - Làm quen với kỹ thuật lập trình hướng đối tượng, hướng sự kiện.
- **Yêu cầu:**
 - Biết và sử dụng các công cụ do Visual Foxpro cung cấp.
 - Thao tác trên CSDL
 - Tìm hiểu các lệnh và các hàm trong Foxpro

Tài liệu tham khảo

- Sử dụng và khai thác Microsoft Visual Foxpro 6.0, Nguyễn Ngọc Minh, NXB Lao Động-Xã hội.
- Microsoft Visual Foxpro 6.0, VN-Guide biên dịch, NXB Thống kê.
- Giáo trình Microsoft Visual Foxpro, Nguyễn Tế An - Nguyễn Tiến Dũng, NXB Giáo dục.
- Quản trị CSDL với Microsoft Visual Foxpro, Văn Thông, NXB Thống kê.
- Giáo trình lý thuyết và bài tập Microsoft Visual Foxpro 6.0, Nguyễn Tiến-Nguyễn Văn Hoài-Nguyễn Văn Tâm, NXB Thống kê.
- Foxpro toàn tập, Seatic.

NỘI DUNG

- Giới thiệu về Visual Foxpro
- Kiểu dữ liệu cơ bản
- Cơ sở dữ liệu
- Query, View
- Form
- Report
- Menu
- Lập trình trong Visual Foxpro

BÀI 1: GIỚI THIỆU

- Cơ sở dữ liệu (Database-DBF) là các tập tin dữ liệu lưu trữ trong bộ nhớ máy tính theo một cấu trúc nào đó
- Hệ quản trị cơ sở dữ liệu (Database Manager System-DBMS) là phần chương trình có thể xử lý, thay đổi dữ liệu
- Quá trình phát triển từ Foxbase – Foxpro for Dos – Foxpro for Windows - Visual Foxpro đều là sản phẩm của hãng Microsoft, là một trong các công cụ nhằm giải quyết các bài toán trong lĩnh vực quản lý

GIỚI THIỆU


- Visual Foxpro 6.0 là:
 - Được tích hợp trong bộ công cụ Visual Studio của hãng Microsoft, dùng để khai thác phần mềm ứng dụng
 - Là hệ quản trị CSDL quan hệ có nhiều công cụ tổng hợp (Table, Query, Form,...), có tính tương tác mạnh
 - Môi trường giúp việc xây dựng cơ sở dữ liệu và phát triển ứng dụng

GIỚI THIỆU

- **Cấu hình tối thiểu của máy tính:**
 - Tốc độ CPU 100 MHz.
 - RAM 16 Mb.
 - HDD 90 Mb cho phần mềm và 60-800 Mb cho thư viện MSDB (Microsoft Developer Network Library).
 - Hệ điều hành từ Windows 95 trở lên.
- Chú ý:** Cài đặt bằng đĩa CD: Chạy tập tin Setup.exe và thực hiện theo sự hướng dẫn

GIỚI THIỆU

- **Khởi động Visual Foxpro:**

- Kích chuột vào biểu tượng của Visual Foxpro trên Desktop 
- Chọn menu Start / Program / Microsoft Visual Studio 6.0 / Microsoft Visual Foxpro 6.0.
- Giới thiệu sơ lược VS Foxpro 6.0

- **Thoát khỏi Visual Foxpro:**

- Tại cửa sổ lệnh Command, nhập lệnh QUIT (Enter).
- Chọn trên thực đơn File/Exit.
- Nhấn tổ hợp phím Atl + F4.

GIỚI THIỆU

- Visual FoxPro có 2 chế độ làm việc:
- **Chế độ hội thoại (interactive):** Là chế độ trả lời từng câu lệnh một của người sử dụng, có 2 hình thức:
 - *Dùng cửa sổ lệnh (Command):* Lệnh được đưa vào cửa sổ lệnh, sau khi ấn Enter
 - *Dùng thực đơn lệnh (Menu):* Lệnh được ban hành bằng cách kích hoạt thực đơn lệnh (menu)
- **Chế độ chương trình (program):** Các câu lệnh có thể tập trung thành một file chương trình nguồn có phần mở rộng .PRG. Để thực hiện, tại cửa sổ lệnh đưa vào câu lệnh: DO < tên chương trình >

GIỚI THIỆU

- **Tập tin và kiểu tập tin**
- **File:** Là tập hợp các thông tin có quan hệ với nhau và có cùng một cấu trúc
- **Các kiểu tập tin:**
 - ***.dbf:** File dữ liệu-dùng để lưu trữ các bảng dữ liệu (data table)
 - ***.idx, *.cdx:** File chỉ mục (index, compact index)-dùng để sắp xếp dữ liệu theo một trật tự cho trước.
 - ***.prg:** File chương trình (program)-tập hợp nhiều lệnh để thực hiện một công việc nào đó

GIỚI THIỆU

- ***.dbc**: File cơ sở dữ liệu-dùng để chứa các bảng dữ liệu (dbf), mối quan hệ giữa các bảng
- ***.scx**: File Form (screen)-dùng để nhập dữ liệu
- ***.vcx**: File thư viện
- ***.frx**: File report-dùng để kết xuất thông tin
- ***.pjx**: File dự án-dùng để tập hợp, theo dõi tất cả các file chương trình (Prg), form (Scr), menu (mnx), report (frx)... nhằm mục đích xây dựng hoàn chỉnh một ứng dụng
- ***.dll**: File thư viện liên kết động

GIỚI THIỆU

- Một dự án (*project*) hoặc một ứng dụng (*application*)
 - **Project (*.pjx)**: là một tập tin dùng để tập hợp, theo dõi tất cả các file chương trình (prg), form (scr), menu (mnx), report (frx),... nhằm mục đích xây dựng một ứng dụng.
 - **Application (*.app)**: là một tập tin tập hợp các thành phần chương trình (prg), form (scr), menu (mnx), report (frx),... được biên dịch chung thành một chương trình duy nhất.

GIỚI THIỆU

- **Tổ chức lưu trữ dữ liệu**

- Dữ liệu của một ứng dụng được lưu trữ trong một hoặc nhiều cơ sở dữ liệu (Database, *.dbc)
- Một cơ sở dữ liệu chứa một hoặc nhiều bảng dữ liệu (Data table, *.dbf) và các mối quan hệ giữa chúng (relationship)
- Trong một bảng dữ liệu chứa nhiều mẫu tin (records)
- Trong một record có thể có nhiều trường (fields)
- Mỗi trường chứa dữ liệu có kiểu được xác định

GIỚI THIỆU

- **Các thành phần cơ bản:**
- **Toán hạng:** là các dữ liệu tham gia vào các phép toán
 - Ví dụ: $\Delta = b^2 - 4 \cdot a \cdot c$ thì $b, 2, 4, a, c$ là các toán hạng.
- **Hằng:** là đại lượng có giá trị không đổi trong thời gian chương trình thực hiện. (Trừ kiểu Memo)
 - *Hằng kiểu số:* như -2.5, 100, 4.14
 - *Hằng kiểu chuỗi:* được đặt trong hai dấu "... " hoặc '... ' hoặc [...], có độ dài tối đa không quá 253 kí tự. Ví dụ: "abc"; tổng hợp', '123',.....
 - *Hằng kiểu ngày:* phải được đặt trong cặp dấu {...}. Ví dụ: {01/01/96}; {}: ngày rỗng.
 - *Hằng logic:* chỉ có 2 giá trị .T. và .F.

GIỚI THIỆU

- **Biến:** dùng để lưu trữ dữ liệu trong quá trình tính toán
- **Quy định đặt tên biến:**
 - dài không quá 10 kí tự, bắt đầu phải là chữ cái
 - không nên đặt trùng tên các từ khoá của Visual FoxPro, có thể viết bằng chữ in hoa hay chữ thường
 - Kiểu của biến là kiểu của giá trị mà nó đang mang.
- **Có 3 loại biến:**
 - **Biến bộ nhớ:** do người sử dụng tạo ra trong bộ nhớ, khi không sử dụng, có thể giải phóng ra khỏi bộ nhớ.
Ví dụ: hsl = 3.12
ngaysinh = {01/01/88}

GIỚI THIỆU

- **Biến hệ thống**: Được tạo ra ngay từ khi khởi động Visual FoxPro. Có tên bắt đầu bằng dấu gạch nối (_) thường được sử dụng trong vấn đề in ấn, người sử dụng không thể giải phóng biến loại này.
- **Biến trường**: Tên các trường trong tập tin CSDL.
- * **Chú ý**: Nếu truy nhập đến chúng nên sử dụng quy cách: **M.<tên trường>**

GIỚI THIỆU

- **Hàm:** Là những đoạn chương trình được viết sẵn nhằm thực hiện một công việc nào đó. Có thể cho ra một giá trị, hoặc cũng có thể chỉ thi hành một việc nào đó mà không cho ra một trị nào.
- **Cú pháp:** Tên hàm(**đối số 1, đối số 2,...,đối số n**)
 - hoặc có thể một hàm mà không có đối số nào cả nhưng phải có () theo sau.
 - Ví dụ: Date (), Sqrt(x),...
- Có 2 loại hàm: Hàm có sẵn và hàm tự tạo

GIỚI THIỆU

- **Biểu thức:** Là tập hợp của một hay nhiều thành phần như hằng, hàm, biến, phép toán, dấu ngoặc tròn và cho ra một giá trị duy nhất. Trị của biểu thức thuộc một trong 4 kiểu: Number, Char, Date, Logic.
- Thứ tự tính toán sau:
 - * Trong () tính trước, ngoài () tính sau,
 - * Phép toán ưu tiên cao tính trước.
 - * Bên trái tính trước, bên phải tính sau.

GIỚI THIỆU

- **Từ khoá:** Là những từ được Visual FoxPro sử dụng vào một mục đích riêng. Thường là những động từ của lệnh thực hiện. Nếu từ khoá có nhiều hơn 4 ký tự thì khi sử dụng chỉ cần ghi 4 ký tự đầu.
 - Ví dụ: MODIFY COMMAND Hello.PRG có 2 từ khoá là MODIFY và COMMAND có thể viết gọn là: MODI COMM Hello.PRG

BÀI 2: CÁC KIỂU DỮ LIỆU

- **1. Kiểu dữ liệu:** V.FoxPro phải chia dữ liệu thành nhiều kiểu dữ liệu khác nhau.
 - Kiểu số (numeric).
 - Kiểu chuỗi (character).
 - Kiểu ngày tháng (date).
 - Kiểu lý luận (logical).
 - Kiểu bộ nhớ (memo).
 - Kiểu tổng quát (general).

Kiểu số (Numeric)

- **Numeric (N)**: chiếm 8 bytes bộ nhớ hoặc tối đa 20 chữ số gồm cả phần nguyên, phần thập phân và dấu chấm thập phân.
- **Float (F)**: Như kiểu Numeric, dùng để biểu diễn số là các số có dấu chấm động.
- **Integer (I)**: dùng để biểu diễn các dữ liệu là số nguyên, chiếm 4 bytes bộ nhớ.
- **Currency (C)**: kiểu tiền tệ. Dùng để biểu diễn tiền tệ trong dữ liệu, tương tự kiểu Numeric nhưng có dấu \$ ở trước giá trị.

Kiểu số (Numeric)

- Một số hàm liên quan:
 - Hàm ABS(nExp): trả về giá trị tuyệt đối của một số. nExp: số hoặc biểu thức số.
 - Ví dụ: $\text{ABS}(-5) = 5$
 - Hàm BETWEEN(n, nLowValue, nHeightValue): Xác định xem giá trị n nằm trong vùng nLowValue và nHeightValue hay không. Kết quả trả về kiểu logic .T. hoặc .F.
 - Ví dụ: $\text{Between}(2,2,4) = .T.$
 $\text{Between}(2,3,4) = .F.$

Kiểu số (Numeric)

- Hàm $\text{Max}(n\text{Exp}1, n\text{Exp}2, \dots)$: Trả về giá trị lớn nhất của dãy. $n\text{Exp}1, n\text{Exp}2, \dots$ là số hoặc biểu thức
- Hàm $\text{Min}(n\text{Exp}1, n\text{Exp}2, \dots)$: Trả về giá trị nhỏ nhất của dãy. $n\text{Exp}1, n\text{Exp}2, \dots$ là số hoặc biểu thức
- Hàm $\text{Mod}(n, m)$: Trả về số dư của n chia cho m
 - Ví dụ: $\text{Mod}(5,2) = 1$

Kiểu số (Numeric)

- Hàm `Round(nExp, m)`: trả về giá trị sau khi làm tròn số `nExp` với `m` số lẻ
 - Ví dụ: `Round(2234.567,2) = 2234.570`
`Round(2234.567,-2) = 2200.000`
- Hàm `Val(cExp)`: ĐỔI chuỗi số `cExp` thành số. Nếu đổi không thành công trả về giá trị là 0.
 - Ví dụ: `Val("15") = 15`
`Val("a15") = 0`

Kiểu chuỗi (Character)

- **Character (C)**: là chuỗi các ký tự thuộc bảng mã ASCII như tên, họ hoặc là số nhưng không có nhu cầu tính toán như số chứng minh, địa chỉ, số phòng, ... Mỗi dữ liệu kiểu chuỗi có độ dài tối đa 255 ký tự (mỗi ký tự chiếm 1 byte trong bộ nhớ).

Kiểu chuỗi (Character)

- Một số hàm liên quan:
 - Hàm Alltrim(Chuỗi): Loại bỏ các khoảng trống trước và sau của chuỗi.
 - Ví dụ: Allt(“ Visual Foxpro ”) = “Visual Foxpro”
 - Hàm ASC (Ký tự): trả về mã ASCII của ký tự
 - Ví dụ: ASC(“A”) = 65
 - Hàm AT(chuỗi 1, chuỗi 2): trả về một số là vị trí đầu tiên của chuỗi 1 xuất hiện trong chuỗi 2
 - Ví dụ: AT(‘tin’,’Lớp tin K27’) = 5

Kiểu chuỗi (Character)

- Hàm CHR(nASC): cho biết ký tự tương ứng với mã nASC.
 - Ví dụ: CHR(97) = “a”
- Hàm CTOD(chuỗi): chuyển đổi biểu thức chuỗi sang kiểu ngày, nếu chuyển đổi không thành công cho giá trị ngày rỗng
 - Ví dụ: CTOD(“15/02/06”) = {15/02/06}
- Hàm DTOC(ngày): Chuyển đổi biểu thức kiểu ngày sang kiểu chuỗi.

Kiểu chuỗi (Character)

- Hàm `Left(chuỗi, n)`: trả về n ký tự kể từ vị trí đầu tiên bên trái chuỗi
 - Ví dụ: `Left('ABCD', 2) = 'AB'`
- Hàm `Right(chuỗi, n)`: trả về n ký tự kể từ vị trí đầu tiên bên phải chuỗi
 - Ví dụ: `Right('ABCD', 3) = 'BCD'`
- Hàm `Len(chuỗi)`: trả về chiều dài của một chuỗi
 - Ví dụ: `Len('ABCD') = 4`

Kiểu chuỗi (Character)

- Hàm Lower(chuỗi): Đổi chuỗi ký tự hoa sang ký tự thường.
 - Ví dụ: Lower('FOXPRO') = 'foxpro'
- Hàm Ltrim(chuỗi): Cắt các khoảng trống bên trái chuỗi.
 - Ví dụ: Ltrim(' FOXPRO ') = 'FOXPRO '
- Hàm Rtrim(chuỗi): Cắt các khoảng trống bên phải chuỗi.
 - Ví dụ: Rtrim(' FOXPRO ') = ' FOXPRO '
- Hàm Upper(Chuỗi): cho kết quả là chuỗi in hoa của chuỗi.
 - Ví dụ: ?Upper ('nguyen van an') → 'NGUYEN VAN AN'

Kiểu chuỗi (Character)

- Hàm Proper(chuỗi): đổi các ký tự đầu mỗi từ thành chữ hoa
 - Ví dụ: Proper('visual foxpro') = 'Visual Foxpro'
- Hàm STR(nExp): Đổi một số thành một chuỗi số
 - Ví dụ: STR(255) = '255'
- Hàm Substr(chuỗi, n, m): cho ra chuỗi con gồm m ký tự kể từ vị trí thứ n của chuỗi.
 - Ví dụ: Substr('Hệ quản trị CSDL 1', 13, 4) = 'CSDL'

Kiểu ngày tháng (Date)

- **Date (D):** Dùng cho những số liệu dạng ngày tháng như ngày sinh, ngày đến,... những dạng ngày tháng bình thường như mm-dd-yy, dd-mm-yyyy,... Độ dài cố định của dữ liệu kiểu ngày là 8 ký tự.
- **Datetime:** như kiểu dữ liệu Date nhưng có thêm phần giờ từ 00:00:00 am đến 11:59:59 pm

Kiểu ngày tháng (Date)

- Một số hàm liên quan:
 - Hàm CDOW(dExp): trả về ngày thứ mấy trong tuần
 - Ví dụ: Cdw({16/02/2006}) = 'Thursday'
 - Hàm Date(): trả về ngày hiện hành có kiểu chuỗi
 - Ví dụ: Date() = {16/02/2006}
 - Hàm DateTime(): trả về ngày, giờ hiện hành
 - Ví dụ: DateTime() = '16/02/2006 02:04:24 AM'

Kiểu ngày tháng (Date)

- Hàm Cmonth(dExp): trả về tên tháng cho bởi biểu thức ngày. Hoặc hàm Month(dExp): trả về tháng cho bởi biểu thức ngày
 - Ví dụ: Cmonth({16/02/2006}) = 'February'
Month({16/02/2006}) = 2
- Hàm Day(dExp): trả về ngày trong tháng cho bởi biểu thức ngày
 - Ví dụ: Day({16/02/2006}) = 16
- Hàm Year(dExp): trả về năm của biểu thức ngày
 - Ví dụ: Year(Date()) = 2006

Kiểu logic (Logical) và Kiểu ghi nhớ (Memo)

- ***Kiểu logic - Logical (L)***: Dùng cho dữ liệu chỉ có một trong hai trường hợp hoặc đúng (T) hoặc sai (F) như giới tính, đối tượng ưu tiên, ... Độ dài cố định của dữ liệu kiểu lý luận là 1 ký tự.
- ***Kiểu ghi nhớ: Memo (M)***: Dữ liệu kiểu ghi nhớ là một đoạn văn bản có độ dài lớn hơn 255 ký tự, như khen thưởng, lý kịch, quá trình công tác,... nội dung lưu trữ trong một tập tin có phần mở rộng là .FPT (FoxPro Text).

Kiểu tổng quát - General

- ***Kiểu tổng quát - General (G)***: Dùng cho những dữ liệu nhúng (OLE) để chứa dữ liệu như hình ảnh, âm thanh,... dùng trong các chương trình "quản lý nhân sự", "nhận dạng",...

CÁC PHÉP TOÁN

- **Phép toán số học:** Được thực hiện trên các dữ liệu kiểu số, gồm các phép toán:

Phép toán	Ý nghĩa	Ví dụ
-, +	Dấu âm và dương	+5, -7
** hay ^	Lũy thừa	5**2 = 5^2
*, /	Nhân, chia	2*5, 5/7
%	phần dư (modulo)	25%4 → 1
+, -	cộng, trừ	10+2, 45-4

- * **Chú ý:** Độ ưu tiên theo thứ tự như trên. Trừ trường hợp các biểu thức trong ngoặc.

CÁC PHÉP TOÁN

- **Phép toán chuỗi:** Dùng xử lý dữ liệu kiểu chuỗi.
 - Phép toán ghép nối (+): ghép 2 chuỗi, kết quả là một dữ liệu kiểu chuỗi.
 - Ví dụ: ‘Trung tâm’ + ‘ Tin học’ ----> ‘Trung tâm Tin học’
 - Phép toán ghép nối (-): ghép 2 chuỗi và di chuyển các ký tự trống ở chuỗi thứ nhất (nếu có) ra cuối chuỗi tạo thành.
 - Ví dụ: ‘Trung tâm ’ – ‘ Tin học’ --> ‘Trung tâm Tin học ’
 - Phép toán \$: kiểm tra chuỗi bên trái có nằm trong chuỗi bên phải không. Kết quả có kiểu logic.
 - Ví dụ: ‘ab’ \$ ‘Abab’ cho giá trị .T. nhưng ‘ab’ \$ ‘AaBb’ cho giá trị .F.

CÁC PHÉP TOÁN

- **Phép toán ngày:** Hai dữ liệu kiểu ngày có thể trừ (-) cho nhau để cho khoảng cách đại số giữa 2 ngày.

Ví dụ: {01/08/2003} - {05/09/2003} -----> - 35

 {01/08/2003} - {05/07/2003} -----> 25

- Một dữ liệu kiểu ngày có thể cộng (+) hay trừ (-) một số nguyên để cho kết quả là một dữ liệu kiểu ngày.

Ví dụ: {01/08/2003} + 10 -----> {11/08/2003}

 {01/08/2003} - 20 -----> {12/07/2003}

CÁC PHÉP TOÁN

Chú ý: Hai dữ liệu kiểu ngày không thể cộng (+) cho nhau. Một số không thể trừ (-) với một dữ liệu kiểu ngày.

Việc diễn tả thứ tự ngày, tháng, năm phụ thuộc vào thời điểm đang ở hệ thống ngày tháng nào.

- (1) *SET DATE FRENCH |AMERICAN| JAPAN*
- (2) *SET CENTURY OFF|ON*: Quy ước năm được biểu diễn theo dạng hai số (mặc định) hay dạng bốn số
- (3) *SET MARK TO <btức C>*: để ấn định ký tự phân cách ngày, tháng, năm là <btức C>. Dùng lệnh *SET MARK TO* để trở về ký tự phân cách mặc định.

CÁC PHÉP TOÁN

- **Phép toán quan hệ**: so sánh hai giá trị của hai biểu thức cùng kiểu, trả về giá trị kiểu logic

Phép toán	Ý nghĩa	Phép toán	Ý nghĩa
<	nhỏ hơn	<>, !	khác
>	lớn hơn	<=	nhỏ hơn hay bằng
=	bằng	>=	lớn hơn hay bằng
==	bằng chính xác		

Ví dụ:

'abc' = 'ab'	&&kết quả .T.
'abc' == 'ab'	&&kết quả .F.

- Trong kiểu logic, Visual FoxPro quy ước: .T.>.F.

CÁC PHÉP TOÁN

- Hai dữ liệu kiểu chuỗi được so sánh dựa theo nguyên tắc: so sánh 2 mã ASCII của 2 ký tự đầu của hai chuỗi, nếu bằng nhau thì so sánh tiếp.
 - Ví dụ: 'ABCD' < 'ABCE' ----> .T. 'a' < 'A' ----> .F.
- Thiết lập môi trường *SET EXACT ON/OFF* (mặc định là *ON*), nghĩa là:
 - Nếu *SET EXACT ON*, lấy chuỗi dài làm chuẩn.
'ABCD' = 'AB' --> .F., vì chuỗi bên phải không còn ký tự so sánh với C
 - Nếu *SET EXACT OFF*, lấy chuỗi bên phải làm chuẩn
'ABCD' = 'AB' --> .T., kiểm tra chuỗi bên trái bắt đầu bằng chuỗi bên phải hay không?

CÁC PHÉP TOÁN

- **Phép toán logic:** Visual FoxPro có 3 phép toán logic: NOT; AND; OR. Trả về giá trị logic
 - NOT hay ! : phủ định của toán hạng theo sau.
 - AND : cho giá trị .T. nếu tất cả toán hạng đều .T.
 - OR : cho giá trị .F. nếu tất cả toán hạng đều .F.
- Ví dụ:
 - $(5 < 7) \text{ and } \text{not}(2 * 4 > 7)$ && kết quả .F.
 - $!(5 < 7) \text{ or } \text{not}(2 * 4 > 7)$ && kết quả .F.