

TIN HỌC ỨNG DỤNG



Hà Văn Sang

Bộ môn: Tin học TC – KT

Khoa: Hệ Thống Thông Tin Kinh tế - Học Viện Tài Chính

Tel: 0982165568

Email: sanghv@hvtc.edu.vn

Website: <http://www.hvtc.edu.vn/sanghv>

CHƯƠNG II



HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU FOXPRO



I. KHÁI NIỆM CHUNG



1. Cơ sở dữ liệu

a. Tệp dữ liệu

- Thông tin về các đối tượng được quản lý thường lưu trữ dưới dạng bảng
- Các bảng thường gồm các dòng và các cột
- Mỗi cột phản ánh một tiêu thức cần quản lý hay còn gọi là thuộc tính của đối tượng
- Mỗi dòng chứa thông tin về tất cả các thuộc tính của đối tượng.
- Giao của dòng và cột là giá trị cụ thể của một thuộc tính

1. Cơ sở dữ liệu



- Một bảng luôn chứa thông tin về một tập hợp các đối tượng cùng loại

Ví dụ: danh sách sinh viên K41

| STT | HỌ TÊN | NGÀY SINH | GIỚI TÍNH | ĐIỂM |
|-----|--------------|------------|-----------|------|
| 1 | Nguyễn Văn A | 01/01/1985 | Nam | 9 |
| 2 | Cao Đức Cẩm | 12/07/1985 | Nam | 8 |
| 3 | Lê Lan | 23/12/1985 | Nữ | 9 |
| .. | ... | .. | .. | .. |

- Khi lưu trữ trên máy, mỗi bảng như vậy được ghi thành một tệp dữ liệu

1. Cơ sở dữ liệu



Vậy có thể hiểu:

Tệp dữ liệu: là tệp chứa thông tin về các thuộc tính của các đối tượng cùng loại được lưu trên máy

- Mỗi đối tượng sẽ được quản lý trên một số thuộc tính nhất định.
- Một bộ thông tin về tất cả các thuộc tính được gọi là một bản ghi dữ liệu (Record).
- Thông tin về một thuộc tính chứa trong một phần của tệp gọi là một trường (Field).

1. Cơ sở dữ liệu



Tập dữ liệu = bảng

Một bản ghi = một dòng

Một trường = một cột

- Một trường được xác định bởi:

- Tên trường
- Kiểu trường
- Độ rộng
- Số chữ số thập phân

1. Cơ sở dữ liệu



Tên trường (Field Name):

Dùng để đặt tên và phân biệt các trường

Quy tắc đặt tên trường trong Fox:

- Tên trường do người dùng đặt
- Gồm các chữ cái, số, gạch nối và không quá **10** kí tự
- Bắt đầu bởi chữ cái
- Không phân biệt chữ hoa, chữ thường

Ví dụ: STT, HoTen, Ngay_Sinh, DIEM

1. Cơ sở dữ liệu



Kiểu trường (Type):

Dùng để xác định kiểu thông tin chứa trong trường

Các kiểu hay gặp trong Fox:

- Kiểu kí tự (character): thường viết tắt là C, gồm các:
 - + Chữ cái
 - + Chữ số không tính toán
 - + Chữ số không tính toán và các kí tự đặc biệt.

Ví dụ: “Hoàng Hùng”, “37/CP”, “K412101”

1. Cơ sở dữ liệu



- Kiểu số (Numeric): thường viết tắt là N, chứa các số có thể tính toán
- Kiểu Ngày (Date): thường viết tắt là D, dùng để lưu ngày tháng năm cụ thể
- Kiểu Logic (Logical): thường viết tắt là L, dùng để lưu các thông tin có hai giá trị trái ngược nhau của một thuộc tính.

.T. là đúng, .F. là sai

Ví dụ: Giới tính nam>< nữ, mua><bán, nhập><xuất

1. Cơ sở dữ liệu



- Kiểu kí ức (Memo): thường viết tắt là M, là kiểu dữ liệu đặc biệt dùng để lưu các văn bản lớn có độ dài không cố định

Ví dụ:

Tiểu sử, lý lịch của một cán bộ

Thông số của một loại máy

1. Cơ sở dữ liệu



Độ rộng (Width):

Dùng để xác định số kí tự tối đa ghi trong trường

Trong Fox:

- Kiểu kí tự (C) ≤ 254 kí tự
- Kiểu số (N) ≤ 20
- Kiểu Ngày (D) = 8
- Kiểu Logic (L) = 1
- Kiểu kí ức (M) ≤ 254 mở rộng 5000

1. Cơ sở dữ liệu



Số chữ số thập phân (Dec):

Đặc trưng riêng của trường số, xác định số chữ số thập phân được dùng

Nếu số chữ số thập phân là 0 thì đó là số nguyên

Một số chú ý:

- Mỗi bản ghi có độ dài không quá 4000 kí tự
- Mỗi tệp có không quá 128 trường
- Tệp dữ liệu của Fox có phần mở rộng là DBF

I. KHÁI NIỆM CHUNG



b. Cơ sở dữ liệu

- Là tập hợp các tệp dữ liệu có quan hệ với nhau để cung cấp một hệ thống thông tin cho một mục tiêu hoặc một bài toán.

c. Cấu trúc tệp dữ liệu

- Là một bảng để liệt kê các thông tin từng trường gồm có tên, loại, độ rộng

I. KHÁI NIỆM CHUNG



2. Hệ quản trị cơ sở dữ liệu

a. Khái niệm

- Là một hệ thống các chương trình dùng để thao tác trên một cơ sở dữ liệu

b. Các thao tác

- Cập nhập dữ liệu
- Truy vấn dữ liệu.
- Lưu trữ dữ liệu
- Kết xuất thông tin

c. Các hệ quản trị cơ sở dữ liệu

- SQL Server, Oracle
- Foxpro, Access ..

II. Hệ QT CSDL Fox



Giới thiệu

- Fox là hệ quản trị cơ sở dữ liệu được sử dụng phổ biến trong các bài toán quản lý kinh tế
- Fox được phát triển từ Foxbase đến Foxpro rồi đến Visual Foxpro theo hướng tương thích đi lên
- Là bộ sản phẩm của hãng Microsoft
- Muốn sử dụng chương trình phải cài đặt, với tệp khởi động là Foxpro.exe, Foxprow.exe, hoặc VFP.exe

II. Hệ QT CSDL Fox



1. Làm quen với Fox

a. Khởi động

-Để khởi động VS Foxpro ta dùng một trong các cách

Cách 1: Kích kép chuột tại biểu tượng VS Foxpro trên màn hình giao tiếp

Cách 2: Kích Start → Programs → mục tương ứng
→ Tên chương trình Fox

Cách 3: Từ RUN chọn đường dẫn tới tệp chương trình rồi nhấn OK

1. Làm quen với Fox

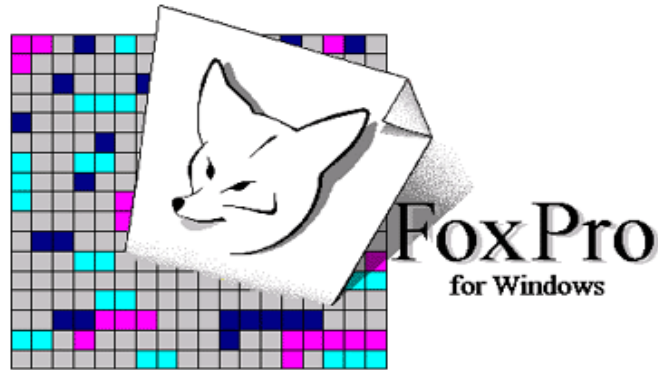
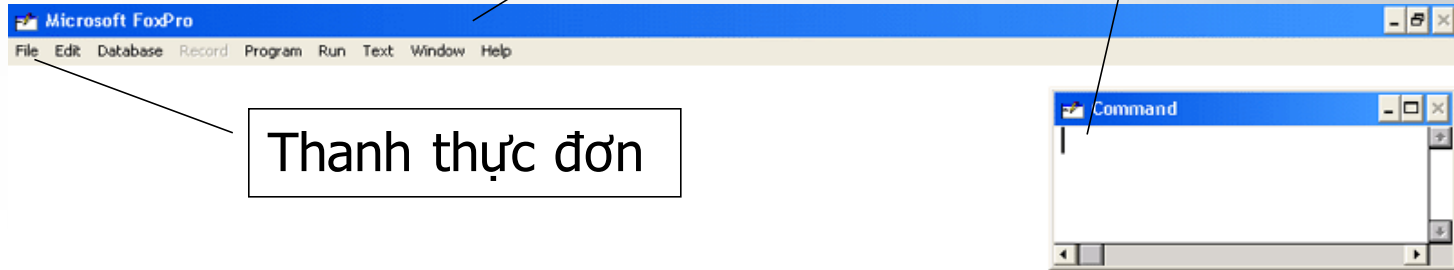


Màn hình chính

Thanh tiêu đề

Cửa sổ lệnh

Thanh thực đơn



FoxPro 2.6 for Windows
(c) 1989-1994 Microsoft Corporation
Product ID xxxxxxxxxxxxxxxxxxxx

1. Làm quen với Fox



Màn hình chính gồm

- Thanh tiêu đề
- Thanh thực đơn
- Màn hình làm việc
- Cửa sổ lệnh (Command window): là nơi vào các lệnh của Fox dưới dạng văn bản

Để bật tắt cửa sổ lệnh ta nhấn Ctrl + F2 hoặc chọn Window → Command

1. Làm quen với Fox



b. Thoát khỏi fox

- Để thoát khỏi foxpro ta thực hiện:

Cách 1: Nhấn tổ hợp phím ALT+ F4

Cách 2: Chọn menu File → Exit

Cách 3: tại cửa sổ lệnh gõ Quit và nhấn Enter

1. Làm quen với Fox



c. Hai chế độ làm việc

Fox làm việc dưới hai chế độ sau:

Chế độ hội thoại (dòng lệnh)

- Là chế độ mà các lệnh được đưa vào từ cửa sổ lệnh, sau khi nhấn Enter lệnh sẽ được thực hiện
- Các lệnh không được lưu trên đĩa nên chỉ dùng để thực hiện các lệnh đơn giản, hoặc các lệnh chỉ dùng 1 lần

1. Làm quen với Fox



Chế độ chương trình

- Là chế độ ghép các lệnh thành một tệp chương trình và ghi lên đĩa với phần mở rộng là .PRG

- Để tạo một chương trình:

Cách 1: dùng một hệ soạn thảo văn bản để viết các lệnh, rồi ghi với phần mở rộng là .prg

Cách 2: tại cửa sổ lệnh gõ:

```
MODIFY COMMAND <tên tệp>
```

Sau đó nhấn Ctrl + W để ghi lên đĩa

- Để chạy một chương trình gõ: DO <tên tệp>

2. Các yếu tố cơ bản của Fox



a. Bộ kí tự

- Bộ kí tự của fox chính là bộ kí tự trong bảng mã ASCII, gồm có:

+ 26 chữ cái a .. z, A .. Z

+ 10 chữ số 0..9

+ Các kí tự +, -, *, /, !, @, #, \$, %

Ngoài ra, visual foxpro có thể sử dụng bảng mã unicode

2. Các yếu tố cơ bản của Fox



b. Từ khóa

- Từ khóa là các từ tiếng anh được Fox qui định với một ý nghĩa xác định nào đó.

Ví dụ:

Các từ khóa lệnh: Modify, Create, List, ..

Các từ khóa phạm vi: next, all, rest ...

Trong fox cho phép chỉ cần viết 4 kí tự đầu của từ khóa.

Ví dụ: Lệnh **Create QLSV** → **Crea QLSV**

Modify Structure → **Modi struc**

2. Các yếu tố cơ bản của Fox



c. Các đại lượng

- Hằng:

Là các đại lượng có giá trị không thay đổi, gồm:

- + **Hằng số**: là các giá trị biểu diễn bởi số 0..9, +, -
- + **Hằng xâu**: là một xâu kí tự bất kỳ có độ dài không quá 255 kí tự.

Hằng xâu được đặt trong:

- Cặp dấu nháy đơn ‘ ’, ví dụ: ‘Tin học’, ‘Hà Nội’
- Cặp dấu nháy kép “”, ví dụ: “Ứng dụng”, “TCKT”
- Cặp dấu ngoặc vuông [], ví dụ: [Học viện]

2. Các yếu tố cơ bản của Fox



+ **Hằng ngày tháng**: chỉ một ngày tháng cụ thể

Cách viết: `{^yyyy-mm-dd}`

Ví dụ:

`{^2006-20-01}`: là sai

`{^2006-10-20}`: là đúng

Để viết được theo kiểu: Ngày-Tháng-Năm dùng lệnh:

SET DATE FRENCH

2. Các yếu tố cơ bản của Fox



+ **Hằng logic**: chỉ giá trị của các mệnh đề logic

Hằng đúng (TRUE): .T. hoặc TRUE

Hằng sai (FALSE): .F. hoặc FALSE

Ví dụ: qui định nam là TRUE, khi đó

Sinh viên nam sẽ có điều kiện:

GioiTinh=.T. hoặc GioiTinh=TRUE

Sinh viên nữ: GioiTinh=.F.

+ Hằng kí Ức:

Giống như hằng xâu

2. Các yếu tố cơ bản của Fox



- **Biến**: là các đại lượng có thể thay đổi giá trị trong quá trình tồn tại, dùng để lưu các giá trị trung gian

Trong Fox có hai loại biến:

+ **Biến trường**

Là tên các trường trong tệp dữ liệu đang làm việc

Nó được tạo ra khi khai báo cấu trúc tệp dữ liệu

Nó tồn tại trên đĩa khi thoát khỏi fox, và chỉ tồn tại trong bộ nhớ khi tệp dữ liệu chứa nó mở

Ví dụ: trường HOTEN, NS, DIEM, LUONG ...

2. Các yếu tố cơ bản của Fox



+ Biến nhớ

Là các biến trung gian xuất hiện trong bộ nhớ trong, dùng để lưu các kết quả trung gian trong quá trình xử lý

Chỉ có 4 loại kiểu biến nhớ: N, C, D, L

Qui tắc đặt tên biến:

- + Gồm các chữ cái, chữ số, không quá 254 kí tự
- + Không bắt đầu bằng chữ số
- + Không đặt trùng tên với biến trường

2. Các yếu tố cơ bản của Fox



+ Để in biến nhớ lên màn hình ta dùng lệnh:

?: in biến nhớ và xuống dòng

?: in biến nhớ và không xuống dòng

Ví dụ:

A="Hà Nội"

Khi đó biến A có giá trị là "Hà Nội", có thể viết A vào các biểu thức, phép so sánh

Lệnh

?A

??A + " Mùa Thu"

Kết quả

Hà Nội

Hà Nội Mùa Thu

2. Các yếu tố cơ bản của Fox



- Hàm:

Hàm là một đoạn chương trình được định nghĩa trước dùng để thực hiện một công việc nào đó và trả về kết quả

Một số hàm hay dùng:

+ **Hàm thao tác trên chuỗi:**

Hàm Len(<bt chuỗi>)

Tác dụng: cho độ dài của biểu thức chuỗi

Ví dụ:

LEN("Hà Nội") = 6 LEN('Tin học') = 7

LEN([Học viện Tài chính]) = 18

2. Các yếu tố cơ bản của Fox



Hàm LEFT(<bt xâu>,<n>)

Tác dụng: trích ra n kí tự bên trái của biểu thức xâu

Ví dụ:

LEFT("ABCD",2)= "AB" LEFT('K41.01.01',3)= "K41"

Hàm RIGHT(<bt xâu>,<n>)

Tác dụng: trích ra n kí tự bên phải của biểu thức xâu

Right("ABCD",2)= "CD" Right('K41.01.01',2)= "01"

2. Các yếu tố cơ bản của Fox



Hàm SUBSTR(<bt xâu>,<m>,<n>)

Tác dụng:

Trích ra n kí tự kể từ vị trí thứ m của biểu thức xâu

Ví dụ:

Substr("ABCD1234",3,4)="CD12"

Substr('K41.01.01',5,2)="01"

Hàm TRIM(<bt xâu>)

Tác dụng: xóa bỏ khoảng trắng bên phải xâu

Trim("AB CD ")="AB CD"

2. Các yếu tố cơ bản của Fox



+ Hàm toán học:

+ Hàm chuyển kiểu:

Hàm VAL(<bt xâu>)

Tác dụng: chuyển xâu thành số

Ví dụ:

VAL("123.45") = 123.45

VAL('12a4h') = 12

VAL([Học viện Tài chính]) = 0

2. Các yếu tố cơ bản của Fox



Hàm STR(<bt số>,[m],[n])

Tác dụng: chuyển giá trị số thành xâu với độ dài m và n số chữ số thập phân

Ví dụ:

STR(123.45) = "123"

STR(123.45,6,2) = "123.45"

STR(20/3,5,2) = " 6.67"

STR(20/3,5) = " 7"

2. Các yếu tố cơ bản của Fox



Hàm DTOC(<bt ngày>)

Tác dụng: chuyển ngày thành chuỗi

Ví dụ:

DTOC({^2006-10-20}) = '10/20/2006'

DTOC({^2006-20-10}) = báo lỗi

Hàm CTOD(<bt chuỗi>)

Tác dụng: chuyển chuỗi thành ngày

CTOD([10/20/2006]) = {10/20/2006}

CTOD('20/10/2006') Báo lỗi

2. Các yếu tố cơ bản của Fox



Hàm DAY(<bt ngày>)

Tác dụng: trả về giá trị của ngày

Ví dụ:

DAY({^2006-10-20}) =20

DAY({^2006}) Lỗi

Hàm MONTH(<bt ngày>)

Tác dụng: trả về giá trị của tháng

MONT({^2006-10-20}) =10

MONT({^2006-20-10}) Lỗi

2. Các yếu tố cơ bản của Fox



Hàm YEAR(<bt ngày>)

Tác dụng: trả về giá trị của năm

Ví dụ:

YEAR({^2006-10-20}) = 2006

YEAR({^2006-20-10}) Lỗi

2006 – YEAR({^1985-10-20}) = 21

YEAR(DATE())-YEAR({^1985-10-20}) = 21

2. Các yếu tố cơ bản của Fox



d. Phép toán và biểu thức

- Biểu thức số

Là các biểu thức nhận giá trị số, gồm các đại lượng số kết hợp với nhau bằng các phép toán

Cộng (+), trừ (-), nhân (*), chia (/), lũy thừa (^)

Đổi dấu, ngoặc ()

Thứ tự ưu tiên:

- Biểu thức trong ngoặc ()
- Phép toán một ngôi (đổi dấu)
- Tính hàm
- $\wedge \rightarrow * \rightarrow / \rightarrow + \rightarrow -$

2. Các yếu tố cơ bản của Fox



- Biểu thức xâu

Là các biểu thức nhận giá trị xâu, gồm các đại lượng xâu kết hợp với nhau bằng các phép toán:

Cộng (+), trừ (-): ghép xâu

Phép (-) sẽ xóa bỏ kí tự trắng ở giữa

Ví dụ

`LEFT("ABCDEF",2) + "EF" = "ABEF"`

`"AB " - "cd" = "ABcd"`

2. Các yếu tố cơ bản của Fox



- Biểu thức ngày

Là các biểu thức nhận giá trị ngày, gồm các đại lượng ngày hoặc số kết hợp với nhau bằng các phép toán:

Phép cộng (+)

Phép trừ (-)

Qui tắc: $\langle D \rangle - \langle D \rangle = \langle N \rangle$

$\langle D \rangle \pm \langle N \rangle = \langle D \rangle$

$\{01/20/2006\} - \{01/10/2006\} = 10$

$\{01/20/2006\} + 12 = \{02/01/2006\}$

$\{01/20/2006\} - 12 = \{01/08/2006\}$

2. Các yếu tố cơ bản của Fox



- Biểu thức quan hệ

Là các biểu thức gồm 2 đại lượng cùng loại kết hợp với nhau bằng các phép toán quan hệ:

Phép bằng: =

Phép khác: <>, #

Phép nhỏ hơn: <

Nhỏ hơn hoặc bằng: <=

Phép lớn hơn: >

Lớn hơn hoặc bằng: >=

Kết quả của các phép so sánh là đúng .T. hoặc sai .F.

2. Các yếu tố cơ bản của Fox



Ví dụ:

.T. > .F. = .T.

3+2 > 6 = .F.

“Anh” > “An” = .F.

“Anh” = “An” = .T.

“Nguyen Van A” = “Nguyen VAn A” = .F.

→

| | | | | | | | | | | | |
|---|---|---|---|---|---|--|---|---|---|--|---|
| N | g | u | y | e | n | | V | a | n | | A |
| N | g | u | y | e | n | | V | A | n | | A |

2. Các yếu tố cơ bản của Fox



$\{^2006-01-01\} < \{^2006-02-01\}$ = .T.

$\{^2006-01-01\} <> \{^2006-02-01\}$ = .T.

$\{^2005-12-12\} > \{^2006-01-01\}$ = .F.

$\{^2006-01-01\} < \{^2006-01-13\}$ = .F.

Khi so sánh biểu thức ngày thì so sánh năm trước, cùng năm so sánh tháng, cùng tháng so sánh ngày

2. Các yếu tố cơ bản của Fox



- Biểu thức logic

Là các biểu thức gồm một tập hợp các đại lượng logic, biểu thức logic kết hợp với nhau bằng các phép toán logic:

Phép phủ định: .NOT.

Phép và logic: .AND.

Phép hoặc logic: .OR.

Trình tự thực hiện:

() → .NOT. → .AND. → .OR.

Ví dụ: .NOT. NS > 1982 => NS <= 1982

NS > 1982 .AND. NS < 2006 => 1982 <= NS <= 2006

| A | B | !A | A and B | A or B |
|-----|-----|-----|---------|--------|
| .F. | .F. | .T. | .F. | .F. |
| .F. | .T. | .T. | .F. | .T. |
| .T. | .F. | .F. | .F. | .T. |
| .T. | .T. | .F. | .T. | .T. |

2. Các yếu tố cơ bản của Fox



- Dạng lệnh tổng quát

Lệnh thao tác với tệp:

LỆNH + TÊN TỆP

Ví dụ:

Create QLSV

Use QLSV

2. Các yếu tố cơ bản của Fox



Lệnh thao tác với dữ liệu:

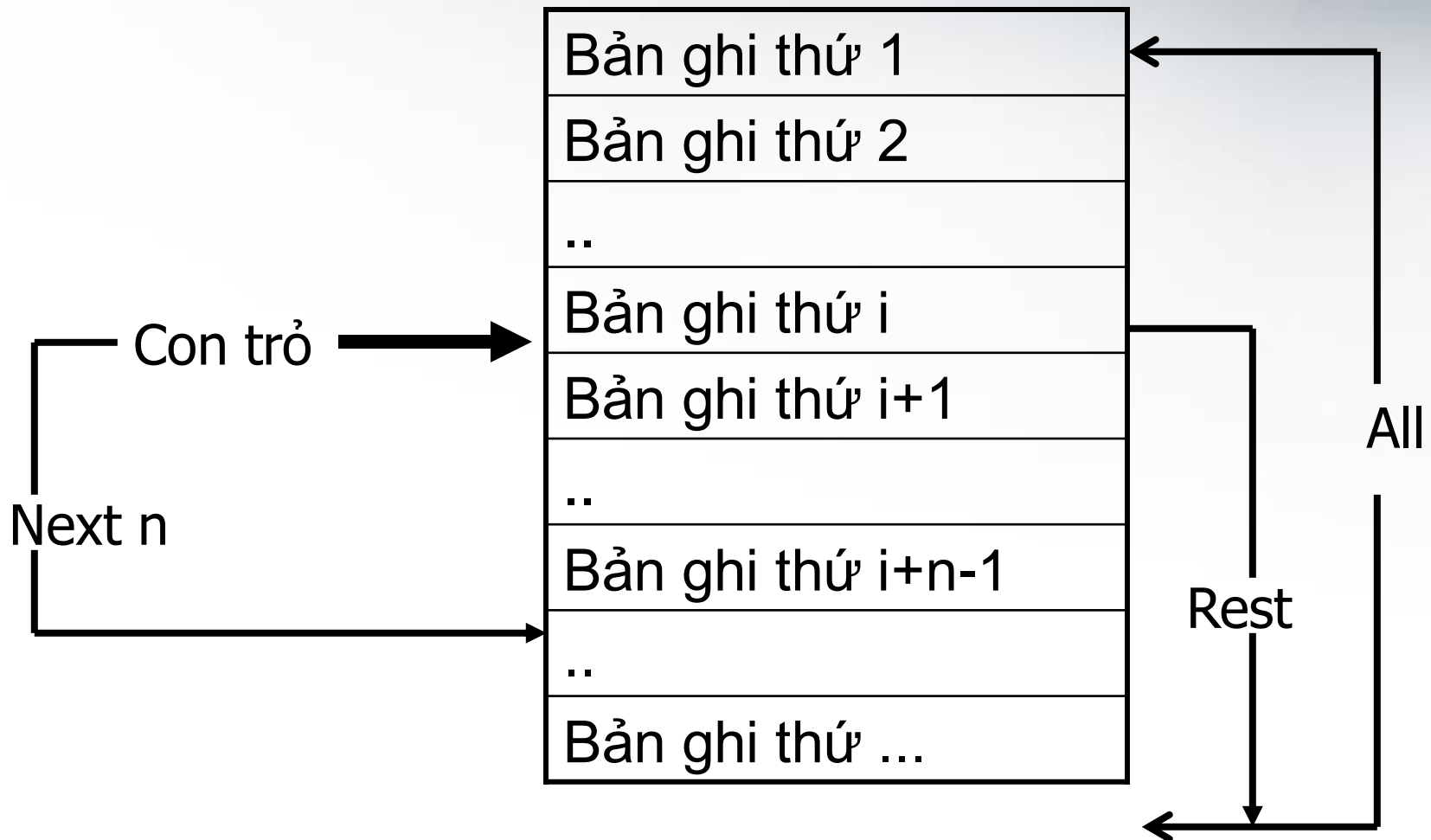
LỆNH [phạm vi] [FIELDS <ds trường>]
[FOR <điều kiện>]

Phạm vi: xác định tập hợp các bản ghi chịu sự tác động

Các giá trị của phạm vi:

- + All: tất cả
- + Next <n>: n bản ghi tiếp theo kể từ bản ghi hiện thời
- + Record <n>: bản ghi thứ n
- + Rest: từ bản ghi hiện tại tới cuối

Sơ đồ mô tả



2. Các yếu tố cơ bản của Fox



FIELDS <Ds trường>: là các trường hoặc các biểu thức chịu sự tác động của lệnh

FOR <điều kiện>: chỉ những bản ghi thỏa mãn điều kiện mới chịu tác động

Điều kiện này là các biểu thức logic

Ví dụ:

Disp all For GT=.F.

Repl all XL with “Gioi” FOR DTB >=8.0

Dele all for LUONG >300

3. Các lệnh cơ bản của Fox



Fox có rất nhiều lệnh, để dễ nhớ ta chia thành các nhóm lệnh sau:

- Lệnh tạo tệp dữ liệu
- Lệnh di chuyển con trỏ bản ghi
- Lệnh xem sửa tệp dữ liệu
- Lệnh thêm bớt bản ghi
- Lệnh sắp xếp và tìm kiếm nhanh
- Lệnh tính toán trên tệp

3.1 Lệnh tạo tệp dữ liệu



a. Lệnh CREATE

- Tác dụng:

Dùng để tạo ra cấu trúc của tệp dữ liệu

Nghĩa là khai báo cho máy biết có bao nhiêu trường và các đặc trưng của trường

- Dạng lệnh

CREATE <TÊN_TỆP>

Trong đó:

+ Tên_tệp: là tên của tệp dữ liệu, có đuôi ngầm định là .DBF

3.1 Lệnh tạo tệp dữ liệu



- Tác động:

Sau khi gõ lệnh và nhấn Enter, máy sẽ hiển thị một bảng cho phép khai báo cấu trúc tệp:

| Tag | Name | Type | Width | Dec | Field |
|-----|------|------|-------|-----|--|
| | | | | | <input type="button" value="Insert"/> <input type="button" value="Delete"/> <input type="button" value="OK"/> <input type="button" value="Cancel"/> |

e:\fpw26\qldien.dbf Fields: 0 Length: 1

3.1 Lệnh tạo tệp dữ liệu



- Để kết thúc việc tạo cấu trúc tệp dữ liệu, nhấn nút OK hoặc nhấn tổ hợp phím Ctrl + W

Khi đó Fox sẽ hỏi có nhập ngay dữ liệu hay không



Yes: mở cửa sổ nhập các bản ghi

No: kết thúc việc tạo cấu trúc tệp và quay trở về cửa sổ lệnh

3.1 Lệnh tạo tệp dữ liệu



- Ví dụ:

Để quản lý sinh viên người ta lưu trữ các thông tin

- Họ tên, ngày sinh, giới tính
- Lớp (2 kí tự đầu chỉ khóa học), điểm thi 2 môn, điểm trung bình

Lệnh:

```
CREATE QLSV
```

Khai báo bảng cấu trúc như sau:

Bảng QLSV.DBF



| Name | Type | Width | Dec | Giải thích |
|-------|------|-------|-----|---------------------|
| HODEM | C | 25 | 0 | Họ tên |
| TEN | C | 7 | 0 | Tên |
| NS | D | 8 | 0 | Ngày sinh |
| GT | L | 1 | 0 | Giới tính (.T./nam) |
| Lop | C | 8 | 0 | Lớp |
| D1 | N | 2 | 0 | Điểm môn 1 |
| D2 | N | 2 | 0 | Điểm môn 2 |
| DTB | N | 2 | 2 | Điểm trung bình |

3.1 Lệnh tạo tệp dữ liệu



b. Lệnh USE

- Tác dụng:

Dùng để đóng hoặc mở tệp dữ liệu

Thông thường tại một thời điểm chỉ có một tệp dữ liệu được mở

- Dạng lệnh

USE [TÊN_TỆP]

Trong đó:

+ Tên_tệp: là tên của tệp dữ liệu, có đuôi ngầm định là .DBF

3.1 Lệnh tạo tệp dữ liệu



- Tác động:

- Nếu không có tên tệp thì đó là lệnh đóng tất cả các tệp đang mở
- Nếu có tên tệp thì máy sẽ mở tệp đó (là quá trình nạp tệp dữ liệu lên bộ nhớ RAM)
- Khi mở tệp mới thì tệp đang làm việc sẽ bị đóng

- Ví dụ: USE QLSV
 USE

Ngoài ra, để đóng tệp có thể dùng lệnh:

CLOSE ALL

CLOSE DATABASE

3.1 Lệnh tạo tệp dữ liệu



c. Lệnh APPEND

- Tác dụng:

Dùng để thêm bản ghi vào cuối tệp dữ liệu

- Dạng lệnh

APPEND [BLANK]

APPEND FROM <tentep> [FOR<btlogic>]

Trong đó:

+ Tên_tệp: là tên của tệp dữ liệu đã có sẵn các bản ghi

+ BLANK: thêm bản ghi trống không có dữ liệu

3.1 Lệnh tạo tệp dữ liệu



- Tác động:

- Dạng 1: dữ liệu do người dùng nhập trực tiếp bằng tay, kết thúc nhấn Ctrl + W
- Dạng 2: dữ liệu được lấy từ một tệp đã có sẵn dữ liệu

Nếu có FOR thì chỉ những bản ghi thỏa mãn bt logic mới được bổ sung

- Ví dụ 1:

Nhập bổ sung danh sách sinh viên vào tệp QLSV

```
USE QLSV
APPEND
```

3.1 Lệnh tạo tệp dữ liệu



- Ví dụ 2:

Tại ổ A:\ có tệp DSLOP.DBF chứa dữ liệu và cấu trúc giống tệp QLSV, yêu cầu sao chép sinh viên khóa 44 vào tệp QLSV

Lệnh:

```
USE  QLSV  
APPEND FROM A:\DSLOP.DBF  FOR LEFT(LOP,2)="44"
```

Chú ý: không dùng lệnh Copy vì nếu copy sẽ ghi đè gây mất dữ liệu đã có.

3.1 Lệnh tạo tệp dữ liệu



d. Lệnh COPY TO
- Xem Giáo trình

3.1 Lệnh tạo tệp dữ liệu



e. Lệnh COPY STRUCTURE

Xem giáo trình

f. Lệnh Copy file

Xem giáo trình

3.2 Lệnh di chuyển con trỏ



a. Khái niệm

- Con trỏ bản ghi:

- + Là cơ chế để định vị bản ghi đang làm việc
- + Tại một thời điểm bản ghi đang được xử lý gọi là bản ghi hiện thời
- + Khi mới mở tệp con trỏ bản ghi nằm ở bản ghi đầu tiên
- + Để làm việc với bản ghi nào đó, ta phải di chuyển con trỏ đến bằng các lệnh sau:

3.2 Lệnh di chuyển con trỏ



b. Lệnh GOTO (GO)

- Tác dụng:

Dùng để di chuyển con trỏ bản ghi tới một vị trí xác định

- Dạng lệnh

GOTO <n>

GOTO TOP/BOTTOM

- n: là số nguyên dương chỉ số hiệu bản ghi cần chuyển đến

- TOP: là từ khóa chỉ bản ghi đầu tiên

- BOTTOM: là từ khóa chỉ bản ghi cuối cùng

3.2 Lệnh di chuyển con trỏ



- Tác động:

- Lệnh sẽ di chuyển con trỏ tới bản ghi có số hiệu n
- Hoặc đến bản ghi đầu tiên nếu dùng TOP
- Hoặc đến bản ghi cuối cùng nếu dùng BOTTOM

- Chú ý:

Hàm Recno() cho biết số hiệu bản ghi hiện tại

Hàm RecCount() cho biết tổng số bản ghi

Nếu $n > \text{RecCount}()$ sẽ báo lỗi

3.2 Lệnh di chuyển con trỏ



Ví dụ:

- Di chuyển tới bản ghi thứ 9 trong tệp QLSV

```
USE QLSV
```

```
GOTO 9
```

```
?RECNO()
```

- Giả sử trong tệp QLSV có 100 bản ghi

Lệnh:

```
USE QLSV
```

```
Go 50
```

```
Go TOP
```

```
Go BOTT
```

Vị trí con trỏ:

1

50

1

100

3.2 Lệnh di chuyển con trỏ



c. Lệnh SKIP

- Tác dụng:

Dùng để di chuyển con trỏ bản ghi tới một vị trí tương đối so với bản ghi hiện thời

- Dạng lệnh

SKIP [n]

- n: là số nguyên bất kỳ

- Nếu không có n thì ngầm định là $n = 1$

3.2 Lệnh di chuyển con trỏ



- Tác động:

- Lệnh sẽ di chuyển con trỏ đi n bản ghi kể từ bản ghi hiện tại
- Nếu $n > 0$: sẽ di chuyển về phía cuối tệp
- Nếu $n < 0$: sẽ di chuyển về phía đầu tệp

- Chú ý:

Nếu con trỏ bản ghi đang ở bản ghi cuối cùng thì khi thực hiện lệnh Skip hàm RECNO() sẽ trả về giá trị là tổng số bản ghi + 1

Hàm EOF() trả về giá trị .T.

3.2 Lệnh di chuyển con trỏ



- Chú ý:(tiếp)

Nếu con trỏ bản ghi đang ở bản ghi đầu tiên thì khi thực hiện lệnh Skip -1, hàm RECNO() sẽ trả về giá trị 1

Hàm BOF() trả về giá trị .T.

- Ví dụ

Giả sử tệp QLSV có 100 bản ghi

| Lệnh: | Vị trí con trỏ: |
|--------------|------------------------|
| USE QLSV | 1 |
| SKIP 5 | 6 |
| SKIP -3 | 3 |
| SKIP | 4 |

3.3 Lệnh xem sửa tệp dữ liệu



a. Lệnh DIR

- Tác dụng:

Dùng để xem các tệp dữ liệu trên đĩa

- Dạng lệnh

DIR

Ví dụ 1 : hiển thị các tệp dữ liệu trong thư mục hiện thời

DIR

Ví dụ 2 : hiển thị các tệp dữ liệu bắt đầu bằng chữ A trong thư mục hiện thời

DIR A*.DBF

3.3 Lệnh xem sửa tệp dữ liệu



b. Lệnh DISPLAY/LIST STRUCTURE

- Tác dụng:

Dùng để hiển thị cấu trúc của tệp dữ liệu đang mở

- Dạng lệnh

```
DISPLAY STRUCTURE [TO PRINTER]  
LIST STRUCTURE [TO PRINTER]
```

3.3 Lệnh xem sửa tệp dữ liệu



- Tác động:

- Khi gặp lệnh này Fox sẽ hiển thị thông tin về cấu trúc tệp dữ liệu đang mở
- Nếu có PRINTER sẽ đưa thông tin ra máy in
- Các thông tin chính gồm có:
 - + Tên trường
 - + Kiểu trường
 - + Độ rộng

Ngoài ra còn có:

- + Số chữ số thập phân
- + Tên tệp đang mở
- + Tổng số bản ghi
- + Ngày cập nhật gần nhất

3.3 Lệnh xem sửa tệp dữ liệu



- Chú ý:

Sự khác nhau giữa hai lệnh List và Display ở chỗ:

- Lệnh List sẽ hiển thị toàn bộ thông tin
- Lệnh Display sẽ hiển thị thông tin theo chế độ phân trang

Ví dụ:

Xem cấu trúc của tệp QLSV

USE QLSV

DISP STRU (Hoặc LIST STRU)

3.3 Lệnh xem sửa tệp dữ liệu



c. Lệnh MODIFY STRUCTURE

- Tác dụng:

Dùng để sửa cấu trúc của tệp dữ liệu đang mở

- Dạng lệnh

MODIFY STRUCTURE

- Tác động

Khi gặp lệnh này fox sẽ hiển thị bảng cấu trúc, cho phép chỉnh sửa, thêm bớt các trường

Nếu chưa có tệp nào được mở, fox sẽ yêu cầu chọn tệp để chỉnh sửa

Bảng cấu trúc tệp



Thêm trường

Xóa trường

Table Structure

| Tag | Name | Type | Width | Dec | Field |
|-----|--------|-----------|-------|-----|-------|
| | hodem | Character | 15 | | |
| | ten | Character | 7 | | |
| | donvi | Character | 4 | | |
| + | luong | Numeric | 6 | 0 | |
| | phucap | Numeric | 6 | 0 | |
| | thuong | Numeric | 6 | 0 | |
| | tamung | Numeric | 6 | 0 | |
| | conlai | Numeric | 6 | 0 | |

Field

Insert

Delete

OK

Cancel

e:\fpw26\bangluon.dbf Fields: 8 Length: 57

Lưu thay đổi

3.3 Lệnh xem sửa tệp dữ liệu



d. Lệnh DISPLAY và LIST

- Tác dụng:

Dùng để hiển thị nội dung các bản ghi của tệp dữ liệu đang mở

- Dạng lệnh

```
DISPLAY [phạm vi] [FIELDS <DS Trường>] [FOR <BT  
LOGIC>] [TO PRINTER]
```

```
LIST [phạm vi] [FIELDS <DS Trường>]  
[FOR <BT LOGIC>] [TO PRINTER]
```

3.3 Lệnh xem sửa tệp dữ liệu



- Tác động:

- Khi gặp lệnh này Fox sẽ hiển thị thông tin của các bản ghi nằm trong phạm vi và thỏa mãn biểu thức logic
- Nếu có PRINTER sẽ đưa thông tin ra máy in
- Các thông tin là các trường có trong ds trường

Chú ý:

+ Nếu không có phạm vi thì với lệnh Display phạm vi ngầm định là bản ghi hiện tại.

Còn với lệnh List ngầm định là toàn bộ các bản ghi

Lệnh Display sẽ hiển thị thông tin theo từng trang màn hình, còn List thì không

3.3 Lệnh xem sửa tệp dữ liệu



Ví dụ:

- Xem bản ghi 50

Use QLSV

Go 50

Display List

- In danh sách sinh viên khóa 42 gồm thông tin: họ tên, ngày sinh, giới tính

Use QLSV

LIST FIELDS Hodem, Ten, Ns, Gt

FOR Left(Lop,2)="42" TO PRINTER

3.3 Lệnh xem sửa tệp dữ liệu



Ví dụ: (Tiếp)

- Xem danh sách sinh viên nữ có điểm trung bình từ 8.0 trở lên

Use QLSV

Display all FOR .NOT. GT .AND. DTB>=8.0

- In danh sách sinh viên sinh tháng 10 năm 1986

Use QLSV

LIST FIELDS Hodem, Ten, Ns

FOR MONT(NS)=10 .AND. YEAR(NS)=1986

3.3 Lệnh xem sửa tệp dữ liệu



e. Lệnh BROWSE

- Tác dụng:

Dùng để sửa chữa nội dung các bản ghi của tệp dữ liệu đang mở (dữ liệu do người dùng nhập bằng tay)

- Dạng lệnh

BROWSE [FIELDS <ds trường>] [FOR <bt logic>]

Trong đó:

- Phạm vi ngầm định là all
- Ds trường: là các trường cần sửa

3.3 Lệnh xem sửa tệp dữ liệu



- Tác động:

- Khi gập lệnh này Fox sẽ hiển thị tệp dữ liệu dưới dạng bảng để người sử dụng sửa nội dung bản ghi
- Nếu có FOR thì chỉ những bản ghi thỏa mãn bt logic mới được hiển thị
- Sửa xong nhấn Ctrl + W để lưu thay đổi

Ví dụ:

+ Nhập điểm thi hai môn của lớp 41.21.01

Use QLSV

```
BROW Fields hodem,ten,d1,d2 FOR LOP="41.21.01"
```

```
^W
```


3.3 Lệnh xem sửa tệp dữ liệu



f. Lệnh Edit

- Tác dụng:

Tương tự BROWSE

Xem giáo trình

3.3 Lệnh xem sửa tệp dữ liệu



g. Lệnh REPLACE

- Tác dụng:

Dùng để thay thế nội dung trường dữ liệu trong tệp đang mở (dữ liệu do máy đưa vào)

Là lệnh duy nhất đưa được dữ liệu vào trường

- Dạng lệnh

```
REPLACE [Phạm vi] <trường 1> With <bt 1>  
      [, <trường 2> With <bt 2>]  
      ...  
      [, <trường n> With <bt n>]  
[FOR <bt logic>]
```

3.3 Lệnh xem sửa tệp dữ liệu



Trong đó:

- Phạm vi ngầm định là bản ghi hiện tại
- Trường 1, 2 ..n là các trường cần thay thế giá trị
- Biểu thức phải có kiểu phù hợp với kiểu của trường

Tác động:

- Khi gặp lệnh này, Fox sẽ thay thế nội dung của các trường với giá trị của các biểu thức tương ứng
- Chỉ có các bản ghi nằm trong phạm vi và thỏa mãn biểu thức logic mới chịu tác động của lệnh

Chú ý:

- Dữ liệu do máy thay thế tự động nên phải cẩn thận khi dùng lệnh

3.3 Lệnh xem sửa tệp dữ liệu



Ví dụ:

Use QLSV

```
REPL ALL DTB WITH 0, PL WITH “ ”
```

USE

- Nhập 10 vào điểm môn 2 của sinh viên có tên Trần Anh

Use QLSV

```
REPL D2 with 10 FOR Ten=“Anh” .AND. Hodem=“Trần”
```

Hoặc

```
LOCATE FOR Ten=“Anh” .AND. Hodem=“Trần”
```

```
REPL D2 with 10
```

3.3 Lệnh xem sửa tệp dữ liệu



Tính điểm trung bình theo công thức

$$DTB = (D1 \times 2 + D2 \times 3) / 5$$

Phân loại:

- Giỏi nếu $DTB \geq 8.0$
- Khá nếu $8.0 > DTB \geq 7.0$
- TB nếu $7 > DTB \geq 5.0$
- Yếu nếu $TB < 5.0$

```
REPL ALL DTB with (D1*2 + D2*3)/5
```

```
REPL ALL PL with "Giỏi" FOR DTB >= 8.0
```

```
REPL PL with "Khá" FOR DTB >= 7.0 .AND. DTB < 8.0
```

```
REPL PL with "Tb" FOR DTB >= 5.0 .AND. DTB < 7.0
```

```
REPL PL with "Yếu" FOR DTB < 5.0
```

3.4 Lệnh thêm bớt bản ghi



a. Lệnh INSERT

- Tác dụng:

Dùng để chèn một bản ghi vào một vị trí bất kỳ trong tệp

- Dạng lệnh

INSERT [BEFORE] [BLANK]

- Tác động:

Lệnh sẽ cho chèn vào tệp 1 bản ghi, cụ thể:

+Nếu có BEFORE thì chèn vào trước bản ghi hiện thời

+Nếu không có BEFORE thì chèn vào sau bản ghi hiện thời

3.4 Lệnh thêm bớt bản ghi



+ Nếu có BLANK sẽ tự động chèn 1 bản ghi rỗng

Nếu là kiểu C → nhận giá trị là các khoảng trống

Nếu là kiểu N → nhận giá trị 0

Nếu là kiểu L → nhận giá trị .F.

Nếu là kiểu D → nhận giá trị ../../....

+ Nếu không có BLANK sẽ chèn bản ghi sau đó cho người dùng nhập dữ liệu

Chú ý: Lệnh này ít được dùng vì máy phải thực hiện ghi chép lại tệp nên tốc độ rất chậm

3.4 Lệnh thêm bớt bản ghi



Ví dụ: chèn 1 bản ghi số 50 vào tệp QLSV

Use QLSV

GO 50

INSERT BEFORE

- Nhập dữ liệu cho bản ghi

Hoặc cách 2:

Go 50

INSERT BEFORE BLANK

REPL Ten WITH “Anh”, Hodem WITH “Trần”

3.4 Lệnh thêm bớt bản ghi



b. Lệnh DELETE

- Tác dụng:

Dùng để **đánh dấu xóa** các bản ghi trong tệp đang mở

- Dạng lệnh

DELETE [Phạm vi] [FOR <bt logic>]

Trong đó: phạm vi ngầm định là bản ghi hiện thời

- Tác động:

Fox sẽ đánh dấu xóa các bản ghi thuộc phạm vi thỏa mãn biểu thức logic

3.4 Lệnh thêm bớt bản ghi



c. Lệnh RECALL

- Tác dụng:

Dùng để hủy việc đánh dấu xóa các bản ghi trong tệp đang mở

- Dạng lệnh

RECALL [Phạm vi] [FOR <bt logic>]

Trong đó: phạm vi ngầm định là bản ghi hiện thời

- Tác động:

Fox sẽ hủy bỏ việc đánh dấu xóa các bản ghi thuộc phạm vi thỏa mãn biểu thức logic

3.4 Lệnh thêm bớt bản ghi



d. Lệnh PACK

- Tác dụng:

Dùng để loại bỏ các bản ghi đã bị đánh dấu xóa ra khỏi tệp

- Dạng lệnh

PACK

- Tác động:

Fox sẽ xóa hẳn các bản ghi bị đánh dấu xóa khỏi tệp, và không thể phục hồi

3.4 Lệnh thêm bớt bản ghi



e. Lệnh ZAP

- Tác dụng:

Xóa bỏ toàn bộ các bản ghi trong tệp đang mở

- Dạng lệnh

ZAP

- Tác động:

Fox sẽ xóa tất cả các bản ghi trong tệp và chỉ để lại cấu trúc của tệp.

Lệnh này tương đương với việc đánh dấu xóa tất cả rồi Pack nhưng tốc độ thì nhanh hơn.

3.5 Sắp xếp và tìm kiếm



- Sắp xếp và tìm kiếm là công việc thường gặp trong quá trình xử lý của máy tính
- Để tìm kiếm được nhanh trước hết ta phải sắp xếp dữ liệu
- Trước khi sắp xếp ta phải xác định được:
 - ✓ Trường cần sắp xếp hay còn gọi là khóa
 - ✓ Sắp xếp theo thứ tự tăng hay giảm
 - ✓ Trường nào được ưu tiên sắp trước
 - ✓ Có phân biệt chữ hoa hay thường không?

3.5 Sắp xếp và tìm kiếm



a. Lệnh INDEX (tạo chỉ mục)

Tệp chỉ dẫn:

- Là tệp được sử dụng để chỉ dẫn cho tệp dữ liệu gốc
- Phục vụ cho công việc tìm kiếm dữ liệu
- Gồm 2 trường: trường chỉ dẫn và số hiệu bản ghi
- Các bản ghi trong tệp dữ liệu có sử dụng tệp chỉ dẫn được sắp xếp theo thứ tự tăng dần của trường chỉ dẫn

3.5 Sắp xếp và tìm kiếm



a. Lệnh INDEX (tạo chỉ mục)

-Tác dụng:

- Dùng tạo ra một tệp chỉ dẫn để sắp xếp tệp đang mở theo thứ tự tăng dần của khóa sắp xếp

- Dạng lệnh:

```
INDEX ON <BT> TO <Tentep> [FOR <bt logic>]
```

Trong đó:

-BT: là biểu thức bất kỳ dùng làm khóa sắp xếp

-Tentep: là tên tệp chỉ dẫn với đuôi ngầm định .idx

3.5 Sắp xếp và tìm kiếm



- Tác động:

Lệnh sẽ tạo ra tệp chỉ dẫn có tên là <tentep> để sắp xếp các bản ghi của tệp đang mở theo thứ tự tăng dần của khóa là BT

Nếu các bản ghi có cùng giá trị khóa sắp xếp thì fox sẽ sắp xếp các bản ghi theo trật tự đã có

Nếu có FOR thì chỉ những bản ghi thỏa mãn bt logic mới được coi là có mặt trong tệp và tham gia sắp xếp

3.5 Sắp xếp và tìm kiếm



- Ví dụ 1:

In danh sách lớp 41.21.01 theo thứ tự Abc của tên

```
USE QLSV
```

```
INDEX ON ten+hodem TO vd1 FOR Lop="41.21.01"
```

```
LIST TO PRINTER
```

```
USE
```

3.5 Sắp xếp và tìm kiếm



- Ví dụ 2:

In danh sách khóa 41 xếp theo lớp, cùng lớp xếp theo điểm trung bình tăng dần

```
USE QLSV
```

```
INDEX ON LOP + STR(DTB,5,2) TO vd2 FOR  
LEFT(Lop,2)="41"
```

```
DISP all TO PRINTER
```

Use

Để sắp xếp DTB giảm dần ta làm như sau

```
INDEX ON – DTB To cdDTB
```

3.5 Sắp xếp và tìm kiếm



- Ví dụ 3:

In danh sách khóa 41 xếp theo lớp, cùng lớp xếp theo ngày sinh tăng dần

```
USE QLSV
```

```
INDEX ON LOP + STR(YEAR(NS),4) +  
STR(MONTH(NS),2) + STR(DAY(NS),2) TO vd3 FOR  
LEFT(Lop,2)="41"
```

```
DISP ALL TO PRINTER
```

Use

Lưu ý: Khi sắp xếp theo một trường thì viết tên trường sau ON, nhiều trường thì biến tất cả thành kiểu xâu để ghép

3.5 Sắp xếp và tìm kiếm



b. Lệnh SORT (Xem giáo trình)

3.5 Sắp xếp và tìm kiếm



c. Lệnh LOCATE

-Tác dụng:

- Dùng để tìm kiếm và định vị con trỏ đến bản ghi thỏa mãn điều kiện nào đó

- Dạng lệnh:

```
LOCATE FOR <BT LOGIC>
```

- Tác động:

Lệnh sẽ duyệt tuần tự từ đầu tệp để tìm bản ghi đầu tiên thỏa mãn biểu thức logic

Nếu thấy sẽ đặt con trỏ tại bản ghi đó

Nếu không thấy sẽ đặt con trỏ tại bản ghi EOF

3.5 Sắp xếp và tìm kiếm



- Ví dụ 1:

Tìm sinh viên có mã là “HTC123”

```
USE QLSV
```

```
LOCATE FOR MASV=“HTC123”
```

```
DISPLAY
```

```
USE
```

Tìm sinh viên nữ sinh ngày 20/10/1985

```
USE QLSV
```

```
LOCATE FOR .NOT. GT .AND. NS={20/10/1985}
```

```
DISPLAY
```

3.5 Sắp xếp và tìm kiếm



d. Lệnh SEEK và FIND

Xem giáo trình

3.6 Lệnh tính toán



a. Lệnh COUNT

-Tác dụng:

- Dùng để đếm số bản ghi trên tệp

- Dạng lệnh:

```
COUNT [TO <Biến> ] [FOR <bt logic>]
```

- Trong đó:

Biến: là biến nhớ dùng để lưu kết quả sau khi đếm

3.6 Lệnh tính toán



-Tác động:

- Lệnh sẽ đếm số bản ghi thỏa mãn biểu thức logic và gửi kết quả vào biến
- Nếu không có TO <biến> thì hiện kết quả lên màn hình nếu ở chế độ SET TALK ON

Để in kết quả lên màn hình ta dùng lệnh:

? <Tên biến>

?? <Tên biến>

3.6 Lệnh tính toán



- Ví dụ 1:

Tính tỷ lệ khá giỏi của khóa 41

```
USE QLSV
```

```
COUNT TO TSO FOR LEFT(LOP,2)='41'
```

```
COUNT TO TGIOI FOR LEFT(LOP,2)='41' .AND.  
DTB>=8.0
```

```
COUNT TO TKHA FOR LEFT(LOP,2)='41' .AND.  
DTB>=7.0 .AND. DTB<8.0
```

```
? 'Tỷ lệ sv giỏi: ', TGIOI/TSO *100, '%'
```

```
? 'Tỷ lệ sv khá: ', TKHA/TSO *100, '%'
```

3.6 Lệnh tính toán



b. Lệnh SUM

-Tác dụng:

Dùng để tính tổng các trường số trên tệp

- Dạng lệnh:

```
SUM [phạm vi] [ds biểu thức] [TO <ds biến>]
      [FOR <bt logic>]
```

- Trong đó:

Phạm vi ngầm định là tất cả các bản ghi

Ds biểu thức gồm các trường số cần tính tổng, ngầm định là mọi trường số

Ds biến là các biến lưu kết quả sau khi tính tổng

3.6 Lệnh tính toán



-Tác động:

- Lệnh duyệt từ đầu tệp tới cuối tệp để tính tổng các trường số trên các bản ghi
- Sau đó gửi kết quả vào các biến
- Trường hợp có FOR thì chỉ có những bản ghi thỏa mãn biểu thức logic mới được tính tổng

-Ví dụ:

Tệp dữ liệu quản lý vật tư có cấu trúc như sau:



| Thuộc tính | Name | Type | Width | |
|-------------|-------|------|-------|------------|
| Loại | Lp | L | 1 | .T. (Nhập) |
| Số phiếu | Sp | C | 5 | |
| Ngày lập | NI | D | 8 | |
| Nội dung | ND | C | 30 | |
| Tên vật tư | TENVT | C | 15 | |
| Mã vật tư | MAVT | C | 6 | |
| Đơn vị tính | DVT | C | 4 | |
| Khối lượng | KL | N | 10 | |
| Đơn giá | DG | N | 10 | |
| Số tiền | ST | N | 12 | |
| Mã kho | MAKH | C | 2 | |

3.6 Lệnh tính toán



- Ví dụ 1:

Tính tổng khối lượng, số tiền nhập xi măng mã “XM30” trong quý I năm 2006

```
USE QLVT
```

```
SUM KL, ST TO TKL, TST FOR LP .AND. MAVT=“XM30”  
.AND. MONT(NL)<4 .AND. YEAR(NL) =2006
```

?’ Tổng khối lượng: ‘, TKL

?’ Tổng số tiền: ‘, TST

3.6 Lệnh tính toán



- Ví dụ 2:

Giả sử có 2 loại đơn vị tính là KG và Tấn chưa chuẩn hóa, yêu cầu tính tổng khối lượng xi măng XM30 còn lại trong các kho

Các bước:

- Đổi đơn vị từ tấn sang kg (khối lượng, đơn giá)
- Tính tổng nhập
- Tính tổng xuất
- Kết quả bằng tổng nhập – tổng xuất

3.6 Lệnh tính toán



- Ví dụ 2:

```
USE QLVT
```

```
REPL DVT WITH "KG", KL WITH KL*1000
```

```
, DG WITH DG/1000
```

```
FOR DVT="TẤN"
```

```
SUM KL TO TN FOR LP .AND. MAVT="XM30"
```

```
SUM KL TO TX FOR .NOT. LP .AND. MAVT="XM30"
```

?Còn lại là: ', Tn-Tx

3.6 Lệnh tính toán



c. Lệnh AVERAGE

-Tác dụng:

Dùng để tính trung bình cộng các trường số trên tệp

- Dạng lệnh:

```
AVERAGE [ds biểu thức] [TO <ds biến>] [FOR <bt logic>]
```

- Trong đó:

Ds biểu thức gồm các trường số cần tính trung bình cộng, ngầm định là mọi trường số

Ds biến là các biến lưu kết quả sau khi tính trung bình cộng

3.6 Lệnh tính toán



-Tác động:

- Lệnh duyệt từ đầu tệp tới cuối tệp để tính trung bình cộng các trường số trên các bản ghi
- Sau đó gửi kết quả vào các biến
- Trường hợp có FOR thì chỉ có những bản ghi thỏa mãn biểu thức logic mới được tính bình quân

-Ví dụ:

Tính khối lượng bình quân của vật tư có mã XM30:

```
USE QLVT
```

```
AVR KL TO KLBQ FOR MAVT="XM30"
```

```
?KLBQ
```

3.6 Lệnh tính toán



d. Lệnh TOTAL

-Tác dụng:

Dùng để tính tổng trên các nhóm con các bản ghi của tệp

Nhóm bản ghi:

- Là các bản ghi thỏa mãn:

1. Liên tiếp nhau

2. Cùng tiêu thức phân nhóm

| MAVT | TENV | KL | ST |
|------|-------|----|----|
| 01 | Sắt | 3 | 5 |
| 02 | Gạch | 5 | 8 |
| 01 | Sắt 6 | 2 | 6 |



| MAVT | TENV | KL | ST |
|------|-------|----|----|
| 01 | Sắt | 3 | 5 |
| 01 | Sắt 6 | 2 | 6 |
| 02 | Gạch | 5 | 8 |

→ Để phân nhóm ta sắp xếp (Index)

3.6 Lệnh tính toán



- Dạng lệnh:

```
TOTAL ON <BT> To <Tep> [FIELDS <Ds trường>]  
[FOR <bt logic>]
```

- Trong đó:

<bt>: là biểu thức bất kỳ dùng làm tiêu thức phân nhóm

<Tep>: là tên của tệp kết quả (.dbf)

<ds trường>: là tên các trường số cần tính tổng, ngầm định là tất cả các trường số có trong tệp

3.6 Lệnh tính toán



- Tác động:

Fox sẽ duyệt các bản ghi từ đầu tới cuối tệp để chia các bản ghi thành các nhóm căn cứ vào:

- + Vị trí trên tệp
- + Tiêu thức phân nhóm

Sau đó tính tổng các trường số trên từng nhóm bản ghi và gửi kết quả của mỗi nhóm vào một bản ghi của tệp dữ liệu kết quả

Trường hợp có FOR thì chỉ bản ghi thỏa mãn mới được tính tổng

3.6 Lệnh tính toán



- Nhận xét:

Tệp ghi kết quả có:

- + Cấu trúc giống tệp ban đầu
- + Số bản ghi bằng số nhóm con
- + Giá trị ở trường được tính tổng là tổng nhóm tương ứng
- + Giá trị ở trường không được tính tổng là giá trị của trường nằm trên bản ghi đầu tiên của nhóm

Vậy khác nhau giữa sum và total là:

Sum tính tổng trên **một nhóm** bản ghi cụ thể

Total tính tổng trên **nhiều nhóm** bản ghi

3.6 Lệnh tính toán



- Ví dụ 1:

Báo cáo tổng khối lượng, tổng số tiền **từng loại** vật tư trong kho

Các bước:

- Sắp xếp vật tư theo từng nhóm → INDEX
- Tính tổng các nhóm → TOTAL
- Mở tệp kết quả
- Hiển thị kết quả

3.6 Lệnh tính toán



```
USE QLVT
```

```
INDEX ON MAVT TO VD1.IDX FOR LP
```

```
TOTAL ON MAVT TO KQ1.DBF FIELDS KL, ST
```

```
USE KQ1
```

```
LIST FIELDS MAVT, TENVT, KL, ST
```

```
USE
```


3.6 Lệnh tính toán



- Ví dụ 2:

Báo cáo tồn kho từng loại vật tư (đơn vị đã chuẩn hóa)

Các bước:

- Gán KL của chứng từ xuất là số âm
- Sắp xếp vật tư theo từng nhóm → INDEX
- Tính tổng các nhóm → TOTAL
- Mở tệp kết quả
- Hiển thị kết quả

3.6 Lệnh tính toán



USE QLVT

REPL KL WITH -KL FOR .NOT. LP

INDEX ON MAVT TO VD2.IDX

TOTAL ON MAVT TO KQ2.DBF FIELDS KL

USE KQ2

LIST FIELDS MAVT, TENVT, KL

USE

3.6 Lệnh tính toán



- Ví dụ 3:

Tính tổng số tiền **từng loại** vật tư trong từng kho của tháng 9 năm 2006

Các bước:

- Sắp xếp vật tư theo từng nhóm → INDEX
- Tính tổng các nhóm → TOTAL
- Mở tệp kết quả
- Hiển thị kết quả

3.6 Lệnh tính toán



```
USE QLVT
```

```
INDEX ON MAVT+MAKH TO VD3.IDX FOR LP .AND.  
MONT(NL)=9 .AND. YEAR(NL)=2006
```

```
TOTAL ON MAVT+MAKH TO KQ3.DBF FIELDS ST
```

```
USE KQ3
```

```
LIST FIELDS MAVT, TENVT, ST
```

```
USE
```