

Indice:

Capitulo 1: Introduccion a tablas DBF

Capitulo 2: Comandos mas usados

Capitulo 3: Manejo de tablas

Capitulo 4: Manejo de registros

Capitulo 5: Organización de registros

Capitulo 6: Mostrar, imprimir y exportar datos

Capitulo 7: Mover puntero y buscar registros

Capitulo 8: Operaciones con campos y registros

Capitulo 9: Manipular múltiples tablas – relaciones

Capitulo 10: Funciones

Capitulo 11: Ejercicios resueltos

Capitulo 12: Ejercicios propuestos

Bibliografia

Capitulo 1

Introducción a tablas DBF

Base de datos es el conjunto de informacion, la cual ha sido organizada y preparada para servir a un proposito especifico. Por Sistema de Gestion de Base de Datos (DBMS) entendemos a la organizacion sistematica de grandes lotes de informacion. La misma informacion esta organizada de diversas formas, de modo que nos faciliten una busqueda, extraccion, listado, reporte o grafico.

RDBMS: Relational Data Base Management System. Software que administra datos relacionales.

BASE DE DATOS: Coleccion de datos relacionales sobre el mismo tema o proposito, donde se puede manejar en conjunto y de forma fácil múltiples objetos, con los que podemos guardar, mostrar, organizar y analizar datos guardados en forma de TABLAS.

TABLA: Objeto que contiene una coleccion de datos sobre un tema particular, almacenados en forma de filas y columnas. Por ejemplo lista de empleados, clientes, proveedores, productos, insumos, ventas, compras, etc.

QUERY: Objeto que presenta una interrogante o define un juego de características a la BASE DE DATOS, que puede involucrar los datos contenidos en varias TABLAS y cuya respuesta es un DYNASET, una coleccion de REGISTROS dinamicos que responden o cumplen las codiciones o características impuestas en el QUERY.

FORMA: Objeto que arregla los datos de una o mas tablas, en forma conveniente para ver muchos CAMPOS a la vez y obtener la mejor presentacion de sus datos en papel o pantalla, presentando subtotales y totales.

CONTROL: Objeto que muestra datos de un campo, variable o expresion con ciertas características especiales, propias del objeto.

OBJETO: Algo que se puede seleccionar y manipular como una unidad.

Hay dos tipos básicos de bases de datos:

1. Jerarquizadas:

Organizan su contenido en base a un modelo de jerarquías que establece relaciones uno a uno y en forma de árbol. Si es necesario hacer relaciones múltiples, será necesario la repetición de datos.

2. Relacionales:

Organiza una base de datos como un conjunto de tablas rectangulares organizadas en filas (llamadas registros) y columnas (llamadas campos). A cada registro se le asigna automáticamente un numero secuencial, es decir un numero de registro y a cada campo un nombre. Cada una de estas tablas se llama relación y constituye un apoyo a la base de datos.

Campo: Unidad básica de información, respecto a determinado registro. Ejemplo: nombre, edad, sueldo, nota etc. (columnas). Pueden tener nombres de hasta 10 caracteres, iniciándose con una letra y no pueden contener espacios.

Pueden ser de varios tipos:

- **Carácter:** Son los que pueden contener hasta 255 caracteres alfanuméricos y normalmente son campos con los que no se hace operaciones. Ej: nombre, dirección, teléfono, # de LE etc.
- **Número:** Usado para introducir números que podrán ser operados posteriormente. El ancho máximo es de 19 dígitos incluido el separador decimal, los dígitos decimales y el signo (+ o -).
- **Fecha:** Para registrar fechas. Siempre de longitud 8.
- **Lógico:** Para registrar un valor lógico como .T. o .F., Si o No etc. Siempre su longitud es de 1.
- **Memo:** Usado para cuando se requiere ingresar mas de 255 caracteres en un campo o para cuando la longitud de campo debe ser variable. Siempre de una longitud de 10 caracteres, pero en realidad los datos de este campo se almacenan en un archivo aparte con la extensión FPT.

Registro: Es un conjunto de información básica dividida en campos, los cuales pertenecen a un elemento o evento. Se almacenan en el orden en que hayan sido introducidos, aunque después podremos ordenarlos como necesitemos. Ejemplo: un alumno, un curso, un empleado, una transacción etc. (filas)

Un buen ejemplo de tabla es un directorio telefónico, los datos se encuentran clasificados por orden alfabético, por numero de teléfono, dirección etc. notar que la misma información puede presentarse organizada bajo diferentes formas, de modo que nos facilite una búsqueda, una extracción o la emisión de un listado, reporte o un gráfico.

Terminos basicos:

Entidad: Persona, lugar, objeto o evento de interes acerca del cual se capturan datos. Ejemplos: Alumnos, cursos, matricula...

Tabla maestro: Contiene datos que describen el estado actual de cierta entidad específica. Siempre debe mantenerse actualizado. Ejemplo: maestro de alumnos, contiene nombre dirección, teléfono y promedio ponderado.

Tabla de transacciones: Es un archivo temporal, acumula datos de cierta entidad respecto a un periodo de tiempo. Ejemplo: matrícula 99-1, contendrá los cursos en los que cada alumno está matriculado para cierto ciclo.

Tabla cabecera: Es un archivo usado para acumular los datos generales pertinentes a un documento. Factura, boleta, guía etc

Tabla detalle: Es un archivo que complementa al anterior, donde están la descripción de los ítems de dicho documento.

Normalización de base de datos:

A través del proceso de normalización se busca simplificar la organización y evitando datos redundantes.

Formas normales:

En primer lugar se deben recolectar todos los datos de la entidad en análisis.

Se busca agrupar datos, de modo que cada grupo represente a una entidad particular, cada grupo representa una tabla. Como consecuencia a cada tabla se le asigna una llave única que los identifique. No se debe incluir campos que no guarden relación con el criterio de grupo. Ej nombre y dirección del cliente en la tabla de transacciones. Tampoco se debe incluir campos en los cuales todos los registros tengan valores iguales.

-

Se procede con la revisión de cada grupo, si en alguno de ellos hubiera datos que no dependan de la llave primaria entonces se sugiere retirarlos.

-

Se sugiere eliminar aquellos datos que se puedan conseguir a partir de otros datos existentes. Ejemplos: Total = Unidades x PU o IGV = 18% de Total.

En la medida de lo posible, pero sin atentar contra la claridad del sistema, se debe tratar de ahorrar espacios en los campos, esto se podrá lograr mediante la codificación de los datos o abreviar los nombres de campos a por menos el ancho del campo. Ej no usar el nombre MATRICULADO para un campo lógico (solo un carácter)

Capítulo 2

Comandos más usados

En los siguientes capítulos se ofrece una descripción general de los comandos y las funciones más potentes y usados de FoxPRO agrupados según la tarea básica que realizan.

Simbología:

La simbología usada en esta separata es la adoptada en la mayoría de los textos de FoxPRO.

<> Indica que el ítem encerrado deberá ser suministrado por el usuario al momento de usar el comando o función. Los símbolos <> no deberán ser incluidos.

<Exp> Puede consistir en un campo, una variable o una combinacion valida de campos variables y funciones. Puede ser caracter, numerico, fecha o logico.

<Var> Se trata de una variable de memoria.

<Condicion>: Expresion que debera responder con verdadero o falso al ser analizado en cada registro.

[] Indican que el ítem encerrado es opcional. Los corchetes no son parte del comando.

| indica que se debe escoger una entre dos o mas opciones.

Alcance: Es una parte opcional de muchos comandos, que indica el grupo de registros sobre el cual se aplicará el comando. El parametros FOR y WHILE establecen por defecto el alcance ALL.

- RECORD n: Un solo registro, el numero n
- NEXT n: Los siguientes n registros comenzando por el activo.
- REST: Desde el registro activo hasta el fin de archivo.
- ALL: Todos los registros de la tabla actual.

Capitulo 3

Manejo de tablas (DBF)

1. Crear tablas

Determinar su nombre y definir su estructura indicando el nombre, tipo y longitud de cada campo.

CREA <dbf / ?> [FROM <archivo>]

2. Usar

Abrir una tabla y activar un índice en determinada área.

USE [<dbf>] [ALIAS <alias>] [IN <area>] [ORDE <tag>] [ASCENDING|DESCENDING] [AGAIN]
[EXCLUSIVE/SHARED] [NOUPDATE]

CLOSE DATABASE = Cierra todas las tablas activas y selecciona como activa el area numero 1.

Algunas funciones útiles:

DBF([<area>|<alias>]) --> da el nombre de la tabla en uso.

ALIAS([<area>]) --> da el nombre del alias de una tabla.

FCOUNT([<area>|<alias>]) --> da número de campos.

FIELD(<#>[<area>|<alias>]) --> da el nombre del campo.

FSize(#) --> Da el tamaño de los campos que corresponden a un registro

HEADER(#) --> La longitud de la cabecera de una tabla

SELECT(0/1) --> Con 0 retorna el # del area activa y con 1 retorna el # del area mas alta no usada aun.

3. Mostrar estructura

LIST STRU [NOCONSOLE] [TO PRIN | TO FILE <txt>]

DISP STRU [NOCONSOLE] [IN <area>] [TO PRIN | TO FILE <txt>]

4. Estructura

MODI STRU = modifica la estructura de la tabla activa

COPY STRU TO <archivo> FIELD <campos> [EXTENDED]

5. Buscar tablas

DIR [ON <d:>][LIKE[<ruta>][<skel>]][TO PRINTER | TO FILE <txt>]

6. Areas de trabajo

FoxPRO puede manejar 225 áreas (solo 25 en versión estándar), en cada una de las cuales puede abrir una tabla y mantener separadamente la posición del puntero.

De forma predeterminada, el área de trabajo número 1 (A) está activa cuando se inicia FoxPRO. Puede especificar un área de trabajo distinta con: SELE <#>. Si <#> es 0, se seleccionará el área de trabajo libre cuyo número sea inferior.

SELECT(0) --> Da el # del área de trabajo activa

SELECT(1) --> Da el # del área de trabajo libre mas alta

USED([<#>]) --> da .T. si área está ocupada o .F. si esta libre.

7. Formas de visualizacion

SET STATUS OFF/on --> Muestra o no la barra de estado

SET HEADER ON/off --> Determina si se usara el encabezado o no en LIST, DISP y CALC

SET DATE TO [DMY / MDY / YMD] --> Especifica el fomato de las fechas

SET CENTURY OFF/on --> Determina si usa 2 (default) o 4 digitos para representar los años.

SET DECIMALS TO <#> --> Especifica el numero de decimales para las expresiones numerias (0 a 18)

SET FIXED

SET MESSAGE TO [<#> / LEFT / CENTER / RIGHT] [WINDOW <nombre>]

SET STICKY ON/off --> Especifica como el mouse mostrara los menus de foxPRO.

SET TALK ON/off --> Especifica si se mostara o no los comandos ejecutados desde la ventana de

comandos.

Capitulo 4

Manejo de registros

1. Añadir registros

APPE [BLANK][FROM<dbf>[FIEL<campos>]][FOR<cond>]]

[DELIMITED [WITH TAB | WITH <delimit> | WITH BLANK]

[TYPE WK1|WK3|WKS|XLS|]

SET CARRY ON/OFF Si esta en ON copia los datos del registro anterior al nuevo registro añadido.

2. Modificar registros

El comando BROW muestra los datos en una ventana, donde se pueden modificar los datos existentes. Para moverse <PgUp>, <PgDn>, <Tab>, <Shift><Tab>, <Ctrl><W> o <CTRL><End> guardar y salir, <Esc> salir sin salvar, <Ctrl><G> añadir registro, <Ctrl><R> marcar/desmarcar para borrar.

BROW [LAST] [FIELDS <campo1,compo2,...>]

[FOR <cond>] [KEY <exp1>,<exp2>]

[VALID [:F] <exp> [ERROR <mensaje>]] [WHEN <cond>]

[FREEZE <campo>] [LOCK <exp>] [TITLE <titulo>][WIDTH <ancho>] [TIMEOUT <#>] [IN WINDOW <ventana>][COLOR <par1>,<par2>...,<par8>]

[NOAPPEND] [NODELETE] [NOEDIT/NOMODIFY] [NOLGRID] [NORGRID] [NOLINK] [NOMENU] [NOWAIT] [NOCLEAR] [NOMENU]

Si se usa la clausula FIELDS, es posible indicar las características de cada campo de la siguiente manera:

:# --> Ancho del campo en caracteres.

:R --> Solo lectura, no será posible modificar. (Read only)

:P=< mascara> --> (Picture)

:B=<min>,<max>[:F] --> Rango en el que debe estar (Between)

:V=<cond> --> Si la condición es :T: el valor es aceptado, sino será rechazado. Solo verifica si hay algun cambio. (Valid)

:E=<mensaje> --> Mensaje de error. (usado con :V)(Error)

:F --> Fuerza la validación al salir del campo (aunque no cambie) (Forced valid)

:W=<cond> -> Si la condicion es falsa no se podra mover el cursos a dicho campo (When)

Campos calculados: Es un campo creado con una expresion, de solo lectura, usando el formato: <nombre> = <exp>

FOR <cond>: Permite filtrar registros, de manera que solo quedaran visibles los que cumplan la condicion.

KEY <exp1>,<exp2>: Permite filtrar registros entre un rango basandose en la llave indice activa.

VALID [:F] <cond> [ERROR <mensaje>]: Evalua <cond> si se efectuo un cambio o si usate :F, si la condicion es verdadera o 1 te deja continuar, si es falsa o 0 envia el mensaje de ERROR. Es una post-condicion.

WHEN <cond>: Evalua <cond> si es verdadera o 1 te permite posicionarte en el registro, si es falsa o 0 se salta al siguiente registro. Es una pre-condicion.

TITLE <exp>: Permite poner un titulo a la ventana

WIDTH <ancho>: Permite limitar el ancho maximo de cada uno de los campos de forma general.

TIMEOUT <#>: Determina cuantos segundos se esperara a que el usuario ingrese algo. Transcurrido dicho tiempo la ventana del BROWSE se cierra automaticamente.

IN WINDOW <ventana>: El browse se abre dentro de la ventana previamente definida por el usuario.

COLOR <par1,par2...,par8>

# Par	Elemento
1	Resto de registros
2	Campo activo
3	Borde
4	Titulo si Browse esta esta activo
5	Titulo si Browse no esta activo
6	Texto seleccionado
7	Registro activo
8	Sombra

CHANGE y EDIT funcionan de manera muy parecida a BROWSE y usan las mismas clausulas

3. Borrar registros

3.1 Marcar

DELETE [ALL / NEXT<#> / RECORD <#> / REST] [FOR<cond>] [WHILE<cond>] [NOOPTIMIZE]

3.2 Desmarcar

RECALL [ALL / NEXT<#> / RECORD <#> / REST] [FOR<cond>] [WHILE<cond>] [NOOPTIMIZE]

3.3 Empaquetar

PACK [MEMO] [DBF]

4. Mostrar o no los registros marcados

Mientras que con SET DELETE OFF se podrá ver los registros marcados para borrar con un rombo. Con SET DELETE ON ya no se mostraran dichos registros.

DELETED() --> Devuelve verdadero si el registro actual está marcado para su eliminación.

5. Borrar todos los registros

ZAP

Capitulo 5

Organización de registros

Los registros de una tabla se pueden ordenar alfabética, cronológica o numéricamente y existen dos tipos de ordenamiento:

Físico: Que consiste en generar una nueva tabla a partir de la tabla activa, con la misma estructura, cuyos registros se encuentran físicamente ordenados respecto a un campo o conjunto de campos llamada llave o clave. Este método solo es recomendable para tablas estables, es decir las que rara vez se modifican sus registros, ya que una modificación obligaría a generar un nuevo archivo.

`SORT ON <campo> TO <tabla> [FOR <cond>] [WHILE<cond>] [ASCE | DESC] [FIELDS [<campos> | LIKE< mascara> | EXCEPT< mascara>]] [<alcance>]`

Lógicos: Que consiste en la generación de un archivo índice, que contiene el orden bajo el cual debe de verse los registros de la tabla. El campo o campos por los que se hace dicho ordenamiento se llama llave o clave. Este archivo índice es pequeño comparado al tamaño de la tabla que contiene los registros y mientras esta activo la modificación o añadido de registros la actualiza automáticamente.

El FoxPRO maneja básicamente dos tipos de índices:

Simples (IDX): Guardan una sola clave, hay que abrirlos manualmente cada vez que se usa la tabla, para que actualice las modificaciones en los registros de la tabla. Usados en FoxBASE.

Compuestos (CDX): Pueden almacenar hasta 25 claves, cada una con una etiqueta (TAG), se abre automáticamente al usar la tabla a la que están asociados, por lo que no hay peligro de desactualizarlo al modificar la tabla y miden aproximadamente 1/6 del tamaño en KB de un archivo IDX. Usados en FoxPRO.

1. Crear

`INDEX ON <campo> TO <idx> | TAG <tag> [FOR <exp>] [ASCENDING | DESCENDING] [UNIQUE] [ADDITIVE]`

2. Abrir IDX

`SET INDEX TO <idx1>,<idx2>,<idx3>`

3. Activar o cambiar

`SET ORDE TO <tag>|<idx> [IN <area>|<alias>] [ASCE | DESC]`

Si usas SET ORDE TO se desactivan todos los índices y quedara la tabla en el orden original, por numero de registro.

4. Reconstruir índices

Reconstruye todos los índices activos.

REINDEX

5. Eliminar índices

DELETE TAG <etiqueta>

6. Poner un filtro

SET FILTER TO <exp>

7. Comandos que afectan los resultados

SET COLLATE TO `spanish`

Afecta la forma como se ordenan los registros (ejemplo: a, b, c, ch, d...) cuando activamos un indice y la forma de compara cadenas, segun el idioma.

? `A' = `a'	.T.	`spanish' o `general'
? `A' = `a'	.F.	`machine'

SET EXACT OFF/on

Afecta la forma de comparar cadenas de diferente longitud.

Si esta en OFF (default) las cadenas se consideran iguales si coinciden caracter a caracter hasta el final de la que esta a la derecha.

Si esta en ON la cadena mas corta se rellena con espacios hasta igualar la longitud de la mas larga.

Ejemplo:

? `Juancho' = `Juan'

Si esta en OFF --> .T.

Si esta en ON --> .F.

Capitulo 6

Mostrar, imprimir y exportar datos

1. List

LIST [OFF][<campo1>,<campo2>...][FOR<cond>]

[WHILE<cond>][TO PRIN | FILE <txt>]

[NOCONSOLE]

2. Display

Muestra registros o expresiones de una tabla.

DISP [ALL][OFF][<campo1>,<campo2>...][FOR<cond>]

[WHILE <cond>][TO PRINTER | FILE <txt>]

[NOCONSOLE]

DISP MEMORY --> Muestra el contenido de las variables de memoria y las matrices.

DISP STATUS --> Muestra el estado del entorno FoxPro.

DISP STRUCTURE --> Muestra la estructura de una tabla.

3. Uno a uno

? | ?? <exp1>,<exp2>,<exp3>...[PICT <exp>][FUNC <exp>][AT <exp>]

??? <exp> --->directo a la impresora

4. Comandos adicionales

SET HEADING ON/OFF

SET FILTER TO <cond> = Usado para hacer visible SOLO a los registros que cumplan con la condicion.

SET FILTER ON/OFF/LOCAL/GLOBAL

Capitulo 7

Mover puntero y buscar registros

1. Mover

.1 Absoluto:

Se pone el numero del registro al que queremos ir o mediante el comando: GO <#> [IN <area> | <alias>]

.2 Relativo:

Se queremos movernos a través de los registros de forma relativa y según el índice activo:

GO TOP | BOTTOM [IN <area> | <alias>]

SKIP [<exp>] [IN <area> | <alias>]

RECNO(<area>|<alias>) --> numero de registro activo

RECCOUNT(<area>|<alias>) --> numero total de registros

2. Búsqueda Secuencial

Búsqueda secuencial es la que partiendo del primer registro de una tabla, comprueba uno a uno si cumplen la condición impuesta y se para en el primer registro que la cumpla.

LOCATE [ALL][NEXT <exp>] [FOR <cond>][WHILE <cond>]

Se usa CONTINUE para buscar el próximo registro. Esta búsqueda puede ser lenta en caso de muchos registros (mas de 1'000,000) si no se usa la tecnología RUSHMORE, basada en el uso de la condición FOR sobre un campo indexado con una expresión básica optimizable del tipo:

<exp índice> <operador> <constante>

donde:

<exp índice> debe coincidir exactamente con el índice activo y no debe contener alias.

<operador> debe ser uno de los siguientes: <, >, =, <=, >=, <>, #, !=

<constante> puede ser una variable de memoria o un campo.

3. Búsqueda en indexados

SEEK <var> --> para buscar en base a una variable o si se usa una cadena deberá estar entre comillas.

FIND <expC> --> para buscar en base a una cadena o valor. No es necesario el uso de comillas.

Nota 1: La expresión de búsqueda no es una condición, sino simplemente el valor a buscar en coincidencia con la clave índice en uso en ese momento. Es indispensable que este indexada la tabla para que funcione alguno de estos comandos.

Nota 2: Es posible consultar el éxito de una búsqueda con FOUND(<area>|<alias>), en caso no tener éxito una búsqueda el puntero queda al final de la tabla, lo compruebas con EOF(<area>|<alias>).

Capitulo 8

Operaciones con campos y registros

1. Operadores Lógicos:

Para agrupar usa (), para negar usa (.NOT. o !), conjunción (.AND.) y disyunción (.OR.)

2. Operadores Aritméticos

Sumar (+), restar (-), multiplicar (*), dividir (/), potenciar (^ o **).

3. Operadores Relacionales

Menor que (<), mayor que (>), igual (=), distinto (# o <>), menor o igual (<=), mayor o igual; (>=), exacto (==).

4. Calculo con campos

Hacer cálculos entre campos sobre un mismo registro.

REPLACE<campo> WITH <exp> [ALL] [FOR<cond>] [WHILE<cond>]

5. Calculo con registros

Hacer cálculos sobre un mismo campo en todos los registros.

CALCULATE <exp> [FOR<cond>][TO<var>]

donde puedes usar: CNT(),SUM(),MAX(),MIN(),AVG(),STD(),VAR(), NPV(), etc

SUM <campo>[FOR<cond>][TO<var>]

AVERAGE <campo>[FOR<cond>][TO<var>]

COUNT [FOR<cond>][TO<var>]

Capitulo 9

Manipular múltiples tablas – relaciones

Como consecuencia de la normalización es probable que para cierto caso tengamos un diseño de base de datos compuesto de varias tablas y es necesario conectarlas para lograr los resultados deseados. Esto se hace relacionando las tablas.

El proceso exige abrir una tabla principal (padre) la cual puede o no estar indexada en un area cualquiera y en otra area diferente una segunda tabla (hijo) la cual deberá estar indexada por el campo que determinara la relación. Luego desde el area donde esta el archivo padre, se emite el siguiente comando:

SET RELATION [TO <campo> INTO <area/alias>] [OFF]

Desde este momento obtendremos una conexión entre estas tablas, de manera que cualquier movimiento en la tabla padre se refleja automáticamente en la tabla hijo y para referirte a un campo de la tabla hijo solo debes anteponer el area o alias delante del nombre del campo de la tabla hijo.

Ej: LIST COD,B.NOMBRE,FECHA

Aquí se hace un listado mostrando el campo COD y FECHA que pertenecen a la tabla padre y NOMBRE que pertenece a la tabla hijo y esta ubicada en el area 2.

Se llama relación uno a uno, si un registro del padre corresponde a un registro en el hijo.

Se llama relación muchos a uno, si varios registros del padre corresponden al mismo registro hijo.

Se llama relación uno a muchos, si un registro en el padre corresponde a muchos registros en el hijo y entonces al posicionarse en cierto registro del padre, solo será mostrado el primer registro correspondiente en

el hijo, de manera que no es posible ver los demás. para solucionar este problema hay que emitir el siguiente comando en el area del padre:

SET SKIP TO <area>

Capitulo 10

Funciones incorporadas

Tablas:

DBF(<area>) --> Muestra el nombre de la tabla activa

USED() --> Determina si tabla esta en area activa o no

ALIAS() --> Muestra el sobrenombre de la tabla

FLUSH() Escribe el contenido del buffer en disco

FILE(archivo) Indica si existe o no un archivo

HEADER() Muestra tamaño en bytes del encabezado de tabla

FLOCK() Boquea la tabla actual

Areas:

SELECT() --> Muestra el area de trabajo activa

RELATION() Muestra la relacion entre dos tablas

TARGET() Muestra el alias de la tabla hija

Indices:

CDX() Muestra el nombre del archivo de indice activo

TAG() Muestra la etiqueta del indice activo

KEY() Muestra la esprecion de indexacion activa

ORDER() Da el nombre del IDX o etiqueta (TAG) activa.

FILTER() Da la condición de filtro

Campos:

FCOUNT(<area>) Muestra el numero de campos

FIELD(n,<area>) Muestra el nombre del campo n

FSIZE() Tamaño del campo

Registros:

DELETE(n) Responde .T. si el registro esta marcado para borrar

BOF(<area>|<alias>) Devuelve verdadero si el puntero de registros se encuentra al principio de una tabla.

EOF(<area>|<alias>) Devuelve verdadero si el puntero de registros se encuentra al final de una tabla.

RECNO(<area>) Numero de registro activo

RECCOUNT(<area>) Numero total de registros de la tabla

RECSIZE(<area>) Longitud en bytes del registro

RLOCK() Bloques registro actual o varios

FOUND() Si encontro despues de LOCA FOR <cond>

SEEK(<var>,<area>) Función avanzada que hace la búsqueda al mismo tiempo que da un .T. si la búsqueda fue exitosa.

Capitulo 11

Ejercicios resueltos

- Abrir la tabla MAEALU en el area 3 con el alias PADRE
- USE maealu IN c ALIAS padre
- Indexamos la tabla en uso por el campo COD y asignamos a este orden el nombre CODIGO.
- INDE ON cod TAG código
- Activar la llave código en forma descendente en la tabla en uso.
- SET ORDE TO código DESC
- Mostrar la llave de índice activa y su etiqueta.
- ? KEY(),TAG()
- Mostrar solo 5 caracteres del campo NOM y el campo COD en modo solo lectura, los campos NOTA y EDAD no tienen ningún condición especial.
- BROW FIEL nom:5,cod:R,nota,edad
- Aquí obligamos a que la NOTA ingresada este en el rango de 0 a 20, si no es así da un mensaje de error. El campo COD es normal.
- BROW FIEL cod,nota:B=0,20:E=`La nota debe ser entre 0 y 20'
- Validar la edad de manera que sea forzosamente mayor de 18 años, de lo contrario indicarlo en un mensaje.
- BROW FIEL cod,edad:V=edad>18:F:E=`Debe ser mayor
- Aquí mostraremos todos los campos de los registros que cumplan la condición impuesta. Los alumnos que tengan menos de 21 años y al mismo tiempo estén aprobados en el curso. Además no será posible añadir ni borrar ningún registro.
- BROW FOR edad<21 .AND. nota>10 NOAP NODE
- Solo muestra los campos mencionados de todos los registros de la tabla, pero solo será posible modificar los apellidos y el nombre de los que estén desaprobados en el curso. La NOTA nunca podrá ser modificada.
- BROW FIEL cod,apepat,apemat,nom,nota:R WHEN nota<11
- Aquí el cursor siempre permanecerá en el campo COD.
- BROW FIEL cod,apepat,apemat,nom,nota FREE cod
- Muestra por pantalla sin numero de registro, los campos COD y NOMBRE, de los registros con nota promedio aprobatoria.

- LIST OFF cod,nombre FOR PROM>10
- Muestra por pantalla con pausa, los nombres de los alumnos con mas de 18 años, ordenados alfabéticamente.

SET ORDE TO nombre

DISP nombre FOR (DATE()-nacio)/365>18

Capitulo 12

Ejercicios propuestos

Crear una tabla con el nombre MAEALU que contenga los siguientes campos: (ingresar por lo menos 5 registros)

- **Nombre Tipo Tamaño**
- COD Caracter 5 Código del alumno
- NOMBRE Caracter 30 Apellidos y nombres
- NACIO Fecha 8 Fecha nacimiento
- PROM Numérico 5.2 Nota promedio
- PAGO Lógico 1 Al Dia o no
-

Crear una tabla con el nombre MAEALU que contenga los siguientes campos: (ingresar por lo menos 30 registros)

- **Nombre Tipo Tamaño**
- COD Caracter 5 Código del alumno
- CUR Caracter 5 Código del curso
- TURNO Caracter 1 M, T o N=Noche
- SECC Caracter 1 Cod de sección
- NOTA Numérico 5.2 Nota final
-

Indexar la tabla MAEALU en los siguientes ordenes: CODIGO del alumno, apellidos y nombre, fecha de nacimiento y nota promedio.

•

Como se donde se encuentra el puntero de registro en cierto momento ?

•

Buscar alguien de apellido PAR...

•

Colocar el puntero en el registro # 4 y visualice el registro completo (todos sus campos)

•

Mueva el puntero dos registros hacia atrás segun el indice activo y márkelo para borrarlo.

•

Borrar los registros cuya nota sea menor que 8.

•

Mostrar por pantalla con pausa los apellidos y nombres y nota promedio de los alumnos que están al Dia en

sus pagos, en orden de edad de mayor a menor.

-

Cuantos registros tiene la tabla abierta en el area 2 ?

-

Crear un archivo de texto que contenga los registros completos pero sin numero de registro, de los alumnos matriculados en este curso, en orden alfabético.

-

Con dos ventanas BROW muestra los cursos en los que esta matriculado cada alumno.

-

Con dos ventanas BROW muestra los alumnos que están matriculados en cada curso-seccion.

-

Saque un promedio de notas de todos los alumnos

-

Cuantos alumnos con promedio aprobado hay ?

-

Cuantos alumnos llevan este curso ?

Bibliografía:

Programación en base a eventos

– Cesar A. Bustamante Gutierrez – UNI

FoxPRO 2.6 para DOS y Windows a su alcance

– José Javier García Badell – McGraw Hill

Programación basica con FoxPRO

– Ramón M. Chordá – Rama

Al dia en una hora en FoxPRO 2.6

– José Carlos Corrales – Anaya Multimedia

Guia practica FoxPRO 2.6

– Alejandro Dominguez – Anaya Multimedia

Diseño de programas: 200 algoritmos y un proyectos

– Julio Vasquez Paragulla – Editorial San Marcos

Microsoft FoxPRO for DOS and Windows, language reference

