

CURSO-TALLER BÁSICO DE



Manual del participante

Facilitador : Dr. José Luis Soria Pérez

Teléfono de contacto: 9612037326

correo drjlsp16@gmail.com

Tuxtla Gutiérrez, Chiapas; enero 2017

INTRODUCCIÓN

El presente documento, para su facilidad de estudio, se encuentra estructurado en 4 unidades de aprendizaje:

UNIDAD I Conceptos básicos, VFP 6.0 e Interface gráfica

UNIDAD II Comandos para mostrar registros, visualizar registros

UNIDAD III Comandos para manipular registros, modificar al interior de los registros

UNIDAD IV Funciones y comandos de configuración

Cada comando cuenta con una descripción de lo que hace, su sintaxis, de manera opcional una pequeña evaluación para recordar las opciones más importantes del comando a trabajar. Por último se tienen ejercicios que el participante deberá realizar

Para la realización de los ejercicios se cuenta con el directorio donde residen las tablas para trabajar **d:\CURSO FOX 2017**

La duración aproximada para el curso-taller es de 15 horas, divididos en 5 sesiones de 3 horas.

OBJETIVO DEL CURSO-TALLER

Al finalizar el curso-taller, el/ la participante, empleará la herramienta de Visual Fox Pro V. 6.0 a través de la solución de problemas hipotéticos, con la finalidad de incrementar la productividad y la calidad del trabajo.

ÍNDICE DE CONTENIDO

I Manipulación de tablas

1.1 Conceptos básicos e interface gráfica

1.1.1. Conceptos básicos.....7

1.1.2. ¿Qué es Visual Fox?.....10

1.1.3. Interface Gráfica.....14

1.2. USE.....16

1.3. CLOSE.....17

1.4 QUIT.....18

1.5 SET DEFA TO.....19

1.6. CREATE.....20

1.6.1. TIPOS DE DATOS.....22

1.7 COPY STRUCTURE.....27

1.8 MODIFY STRUCTURE.....28

1.9 IMPORT/EXPORT.....30

II Visualizar registros

2.1. BROWSE.....	39
2.1.1 ALCANCE.....	42
2.2. SET FILTER.....	44
2.3 LOCATE.....	45
2.4. COUNT.....	46
2.5 GO / GO TO.....	48
2.6 SKIP.....	49
2.7 RECNO().....	49
2.8 DELETED().....	50

III Modificar registros

3.1. REPLACE.....	53
3.2. DELETE.....	55

3.3. PACK.....	57
3.4. ZAP.....	58
3.5 RECALL.....	59
3.6 INSERT BLANK	61
3.7 APPEND BLANK.....	62
3.8 APPEND FROM.....	63
3.9. SORT.....	66

IV Funciones

4.1. Comando ? y expresiones.....	72
4.2. Matemáticas	
4.2.1. SUM ().....	72
4.2.2. AVG ().....	74
4.2.3. VAL().....	75

4.3. De cadena

4.3.1. Operador +	76
4.3.2. SUBSTR().....	76
4.3.3 LEN().....	78
4.3.4 LIKE ().....	79
4.3.5 LOWER().....	80
4.3.6 UPPER ()	81
4.3.7 STR ().....	82

4.4. Comandos de configuración

4.4.1.SET STATUS.....	83
4.4.2 SET CARRY.....	84
4.4.3 SET DELETE.....	85
4.4.5 SET EXCLUSIVE.....	86

UNIDAD I

Manipulación de Tablas

INTRODUCCIÓN A LA

UNIDAD I

En la Unidad I se revisarán conceptos básicos sobre las tablas y bases datos.

Por otro lado, se hará un análisis sobre la importancia, características e historia del Visual Fox Pro.

También veremos las distintas formas de cómo podremos dar las instrucciones en la interface: Línea de comandos, menú de opciones, archivo de comandos (prg), asistentes.

Otro aspectos básicos a revisar son el cómo abrir una tabla, cómo cerrarla y salir de Fox con QUIT

Otro tema de vital importancia, es indicarle a FOX la ruta donde se guardarán los archivos con SET DEFAULT

Dado que la tabla es la unidad básica de trabajo, revisaremos cómo podemos crear una (CREATE) y los distintos tipos de campo

El trabajo de modificación de estructura (MODY STRU) y copiado de la estructura (COPY STRU), aunque ocasional, puede ser de gran relevancia.

Por último, se revisará el procedimiento para que podamos recibir tablas de otros formatos (IMPORT) o cómo enviar de Fox a otros formatos (EXPORT).

1.1.1 Conceptos básicos

Base de datos

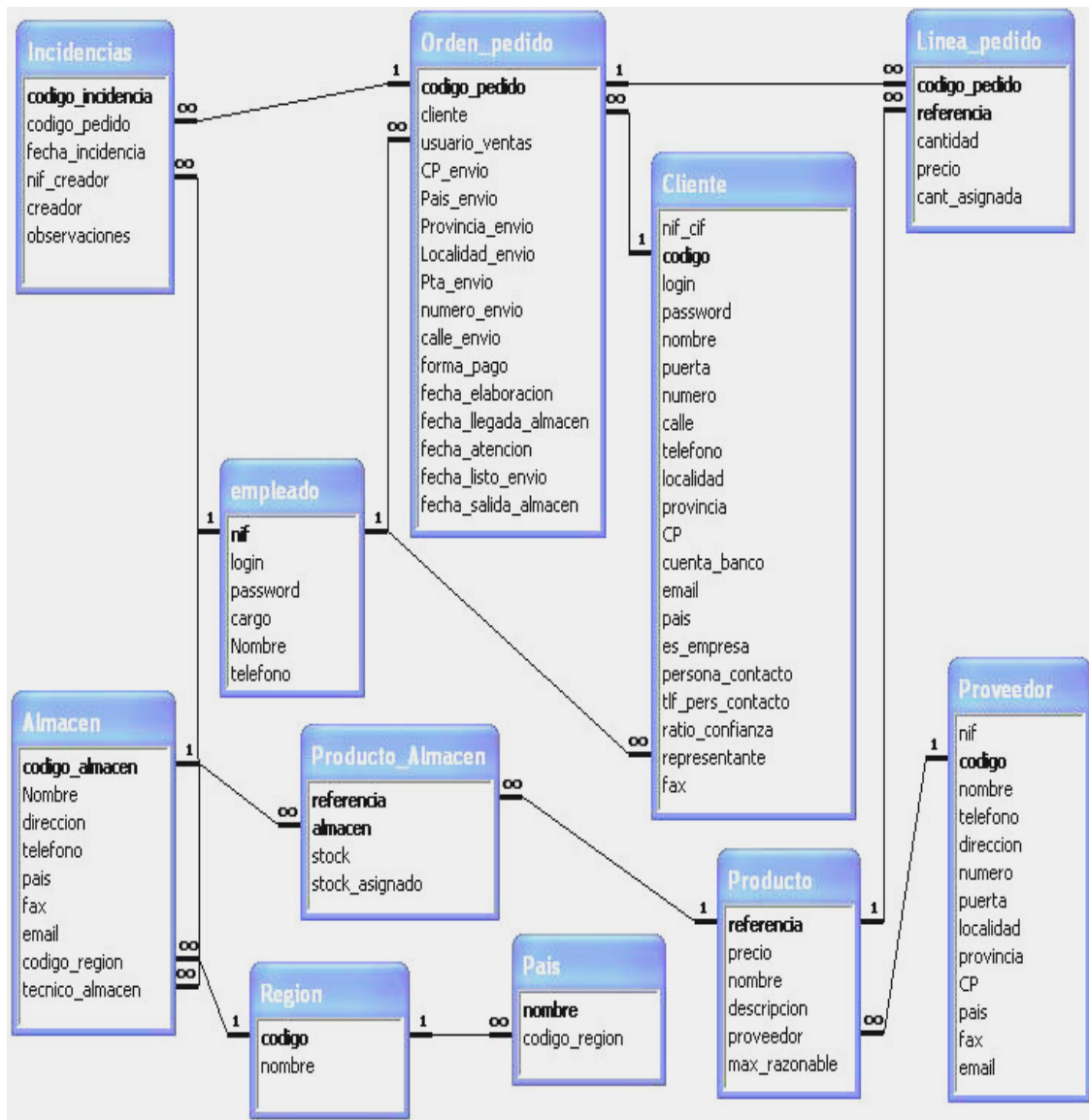
Una base de datos es una colección de archivos relacionados con la finalidad de permitir el manejo de la información de alguna compañía. Cada uno de dichos archivos pueden ser vistos como una colección de registros y cada registro está compuesto de una colección de campos. Cada uno de los campos de cada registro permite llevar información n de alguna característica o atributo de alguna entidad del mundo real.

Base de Datos

- Si hablamos de una UNIVERSIDAD, las entidades de interés podrían ser: ALUMNOS, PROFESORES, SALONES, etc. Respecto a la entidad ALUMNO, los atributos o características importantes podrían ser: NOMBRE, DIRECCION, TELEFONO, CARRERA, MATRICULA, etc.

<http://www.tamps.cinvestav.mx/~jtj/courses/dbs/slides/chapter1dbs.pdf>

Ejemplo de Base de datos



Ejemplo de tabla

Nombre	Teléfono
Vargas Rubén	18-20-67
Díaz Raúl	13-67-90
Ruiz Pedro	12-89-90
Martínez Raúl	12-54-46

Cuadro 1: Tabla para llevar una agenda telefónica

Componentes básicos de una tabla:

Campo 1		Campo 2		Campo N			
Employee I	Last Nam	First Nam	Title	Title C	Birth Date		
+	1	Davolio	Nancy	Sales Representative	Ms.	08-dic-1968	Registro 1
+	2	Fuller	Andrew	Vice President, Sales	Dr.	19-feb-1952	Registro 2
+	3	Leverling	Janet	Sales Representative	Ms.	30-ago-1963	
+	4	Peacock	Margaret	Sales Representative	Mrs.	19-sep-1958	
+	5	Buchanan	Steven	Sales Manager	Mr.	04-mar-1955	
+	6	Suyama	Michael	Sales Representative	Mr.	02-jul-1963	
+	7	King	Robert	Sales Representative	Mr.	29-may-1960	
+	8	Callahan	Laura	Inside Sales Coordinato	Ms.	09-ene-1958	
+	9	Dodsworth	Anne	Sales Representative	Ms.	02-jul-1969	Registro N

1.1.2. ¿Qué es el Visual Fox Pro?

Visual FoxPro es un lenguaje de programación procedural, que posee un Sistema Gestor de Bases de datos o Database Management System (DBMS) y un Sistema Administrador de Bases de Datos Relacionales, producido por Microsoft.

Los DBMS este nombre se conoce a productos de fabricantes como Oracle, Sybase, Informix, Ingres, Borland, Microsoft, IBM, etc.

Dbase y Paradox son programas de base de datos con un DBMS, un lenguaje completo de programación y un lenguaje de cuarta generación, haciendo de ellos sistemas completos de desarrollo de aplicaciones. Los comandos de los lenguajes de cuarta generación permiten a los usuarios crear en forma interactiva archivos de bases de datos, editarlos, formular preguntas e imprimir informes sin necesidad de programación. Miles de aplicaciones han sido desarrolladas en ambientes como éstos.

... Fuente <http://www.mastermagazine.info/termino/4544.php>

Características

Visual FoxPro provee a los desarrolladores un conjunto de herramientas para crear aplicaciones de bases de datos para el escritorio, entornos cliente/servidor, tablet PC o para la Web.

Entre sus características se pueden enumerar:

- Capacidades para el manejo de datos nativos y remotos.
- Utilización de Sentencias SQL.
- Manejo de vistas y estructuras relacionales.
- Su propio gestor de base de datos incorporado. Interconexión con servidores de base de datos, tales como Oracle, Microsoft SQL Server o MySQL
- Cuenta con un motor de generación de informes renovado y flexible para soluciones más robustas.
- Desde la versión 9.0, amplio soporte de XML, tanto como fuente de datos (por ej., servicios Web basados en XML) como por generar reportes en formato XML.

Historia

Visual FoxPro proviene de FoxPro, que a su vez deriva de FoxBASE, creado por Fox Software en 1984; inicialmente un clon de dBase, acabó superándolo y, junto a Clipper, convirtiéndose en una de las estrellas de los lenguajes xBase.

Cuando se presentó FoxPro 2.0 se incluyeron varias tecnologías que revolucionaron el mercado de desarrollo de bases de datos las cuales son:

- La adición de Tecnología Rushmore hizo posible que tablas tuvieran millones de registros sin la necesidad de pasar a tecnologías más caras
- Las instrucciones SQL (Lenguaje Estructurado de Consulta) que reemplazaban procedimientos completos. SQL fue, y todavía es, el lenguaje de los datos

Versiones con Soporte Windows						
Versión	VFP 3.0	VFP 5.0	VFP 6.0	VFP 7.0	VFP 8.0	VFP 9.0
Windows 3.x	Si	No	No	No	No	No
Windows NT 4.0	Si	Si	Si	Si	Si	No ¹¹
Windows 95	Si	Si	Si	Sólo ejecutable	No	No
Windows 98	Si	Si	Si	Si	Sólo ejecutable	Sólo ejecutable
Windows Me	Si	Si	Si	Si	Sólo ejecutable	Sólo ejecutable
Windows 2000	Si	Si	Si	Si	Si	Si
Windows XP	Si	Si	Si	Si	Si	Si
Windows Server 2003	Si	Si	Si	Si	Si	Si
Windows Vista	Modo de compatibilidad	Si	Si	Si	Si	Si
Windows 7	No	Si	Si	Si	Si	Si
Windows 8	No	No	Si	Si	Si	Si
Windows 8.1	No	No	Si	Si	Si	Si
Windows 10	No	No	Si	Si	Si	Si

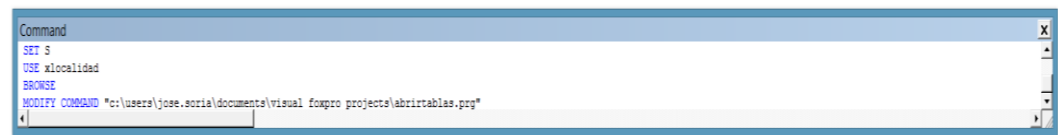
En el ámbito de la informática las nuevas tecnologías reemplazan las anteriores, se incrementa la rapidez de los cambios, por lo que el factor de obsolescencia es alto.

Este cuadro indica que a pesar de la evolución de las versiones de Windows, partiendo de Windows 2000 han podido correr las diferentes versiones del VFP a partir de la versión 6.0

1.1.3 Interface gráfica del VFP6

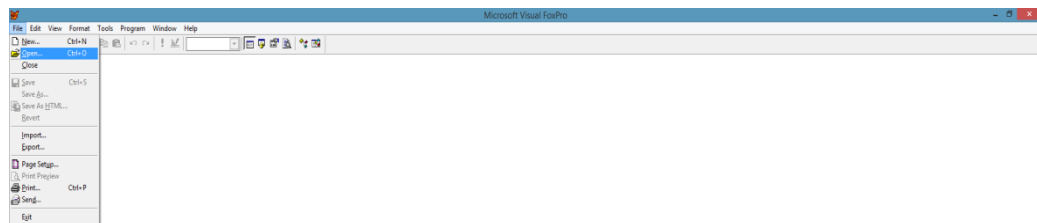
El usuario puede dar las instrucciones desde diferentes elementos de la interface:

- **Línea de comandos**



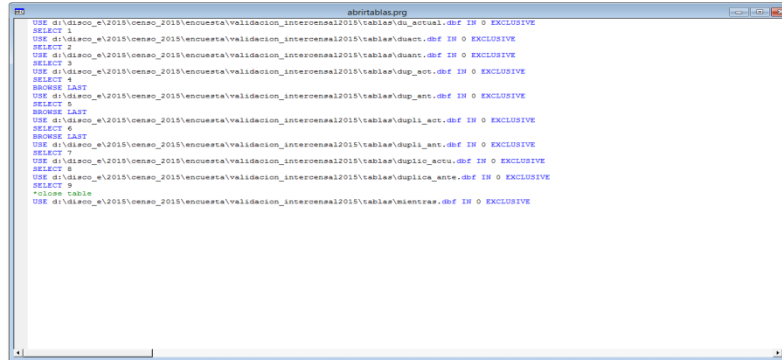
Es válido escribir por lo menos 4 caracteres del comando, por ejemplo: SET DEFA TO es equivalente a SET DEFAULT TO.

Menú de opciones



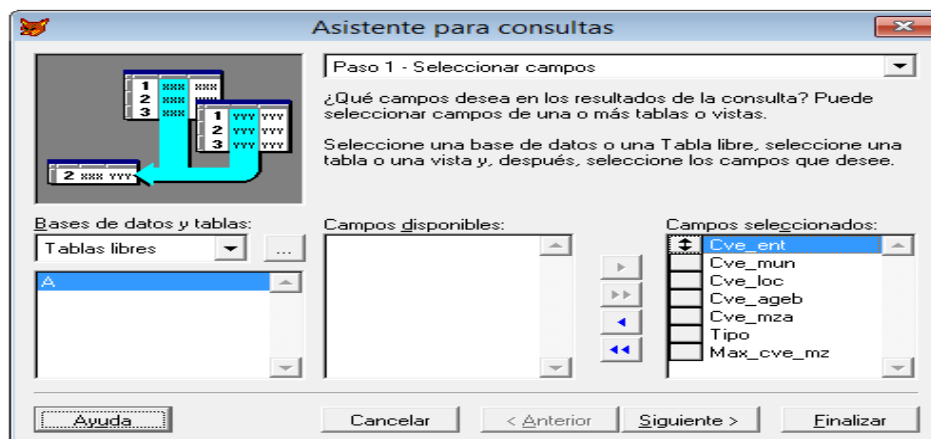
Seleccionar cualquiera de las categorías y posteriormente las opciones disponibles.

- **Editor de archivos PRG**



Permite al usuario crear sus propios programas para la automatización de procesos.

- **Asistentes**



Facilita la creación de tablas, consultas, formularios, reportes para el usuario.

1.2. USE (Comando)

Abre una tabla

Sintaxis

USE [Nombre_de_Tabla] [EXCLUSIVE] [SHARED]

Parámetros

Nombre_de_Tabla

Especifica el nombre de la tabla que se va a abrir.

EXCLUSIVE

Abre la tabla para uso exclusivo en una red.

SHARED

Abre una tabla para uso compartido en una red. SHARED permite abrir una tabla para uso compartido, incluso cuando EXCLUSIVE está establecido en ON.

Abrir una tabla de uso exclusivo garantiza que el archivo no puede ser cambiado por otros usuarios. Para algunos comandos, la ejecución no es posible hasta que se abra una tabla para uso exclusivo. Estos comandos son INSERT, INSERT BLANK, MODIFY STRUCTURE, PACK, REINDEX y ZAP.

Ejercicios:

1. Abrir de forma compartida el archivo ESTADOS.DBF (Shared)
2. Modificar la estructura del archivo
3. Hacer lo mismo con el modo exclusivo

1.3. CLOSE (Comandos)

Cierra archivos

Sintaxis

CLOSE [ALL] [TABLES]

Parámetros

ALL

Cierra todas las bases de datos, tablas e índices abiertos en la sesión de datos actual

TABLES

Cierra todas las tablas de la base de datos seleccionada actualmente.

Ejercicios:

1. Abrir el archivo ESTADOS, de forma exclusiva
2. Cerrar el archivo

1.4 QUIT

Termina la sesión actual de Visual FoxPro y devuelve el control al sistema operativo.

QUIT

Observaciones

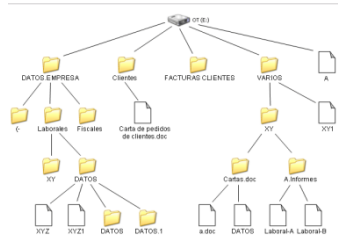
Para finalizar una sesión de Visual FoxPro, debe utilizar siempre el comando QUIT.

Si apaga el equipo sin ejecutar QUIT, se pueden dañar los archivos abiertos y perder datos. Además, puede dejar en el disco archivos temporales de trabajo que normalmente se eliminarían.

1.5. SET DEFAULT (Comando)

El directorio predeterminado es aquél desde el cual se inicia Visual FoxPro , un ejemplo es la ubicación C:\Users\jose.soria\Documents\Visual Foxpro. Es decir, se guarda la ruta donde fue instalado.

Sin embargo, para evitar que los archivos que generemos se queden en ese lugar debemos especificar al VF cuál es la ruta en donde se trabajará por default.



Cambia la ruta de trabajo

Sintaxis

SET DEFAULT TO [cRuta]

Parámetros

cRuta

Ejemplos

Puede especificar un directorio determinado:

SET DEFAULT TO D:\2017

SET DEFAULT TO D:\2017\MGN\PROYECTOS

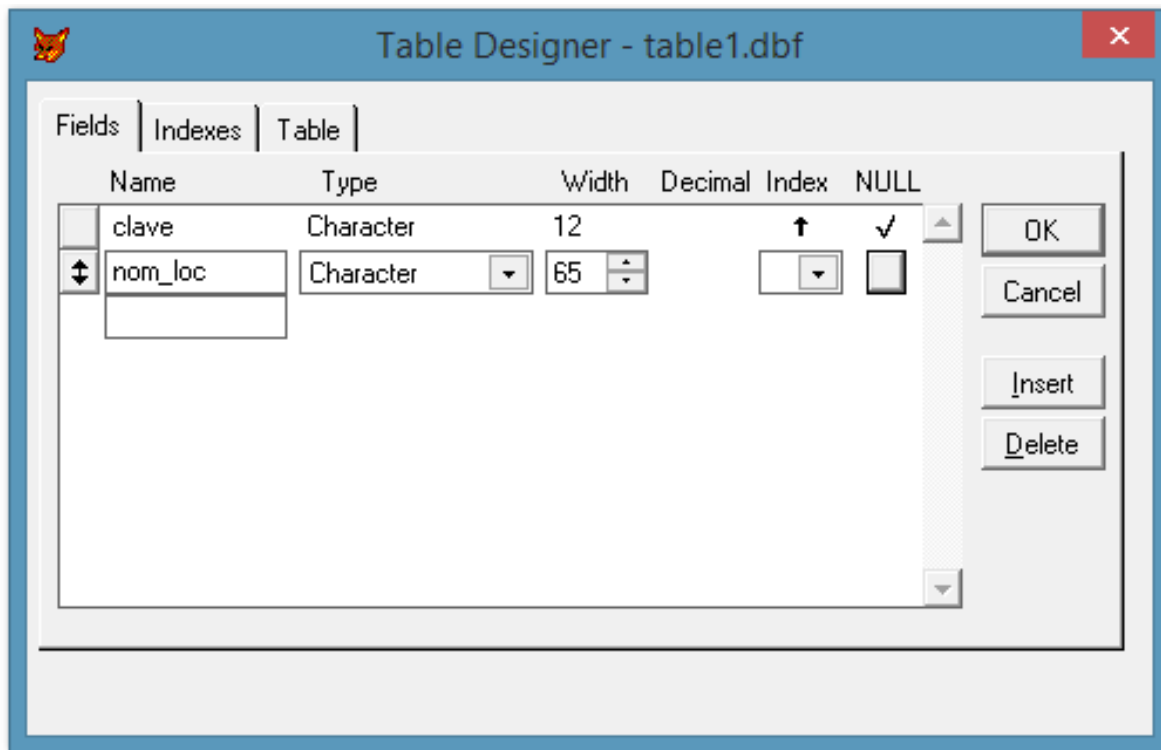
Ejercicios:

1. Establecer como predeterminada la ubicación del D:\Curso de Fox 2017

1.6 CREATE

Aunque una de las formas de crear una nueva tabla es utilizando el asistente, veremos la forma de crearla de manera personalizada, puesto que es la más empleada en la práctica.

Lo primero que debemos hacer es diseñar la tabla o revisar la estructura que nos indican debe tener nuestra tabla



A continuación procederemos a capturar los datos correspondientes a la estructura de la tabla: Name es el nombre del campo, Type es para seleccionar el tipo de datos que tendrá el campo, Width es el ancho del campo. Decimal se utiliza para cuando es numérico y tiene punto decimal, Index si es un campo índice (generalmente va desactivado), NULL si se activa quiere decir que nos se aceptan valores nulos, por ejemplo "" o el valor cero.

Ejercicios:

Crear una tabla con la estructura siguiente:

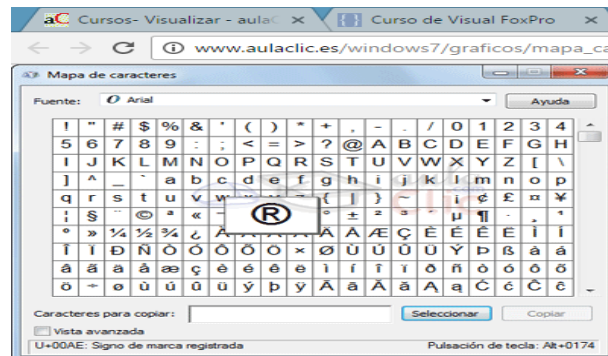
NOMBRE	TIPO	DESCRIPCION
CLAVE	C(12)	CLAVE
MANZANAS	N(3)	CANTIDAD DE MANZANAS
FECHA_ING	D	FECHA DE INGRESO

Esta tabla tendrá el nombre de UNO.DBF

1.6.1. TIPOS DE DATOS DE VISUAL FOX PRO

Los principales tipos de datos se describen brevemente a continuación:

Character Estos pueden usarse para almacenar todo tipo de caracteres, incluyendo letras, números, símbolos especiales y espacios en blanco. El ancho máximo de este tipo es de 254 caracteres.



Numeric Estos se usan para almacenar números, con o sin cifras decimales, así como los signos + y -. El ancho máximo para este tipo es de 20 caracteres y tanto el punto decimal como los signos ocupan un lugar.

Sugerencia: Reserve los campos numéricos para cuando su contenido tenga que ser objeto de cálculo.



Date

Estos campos se usan para almacenar fechas numéricas separadas por una diagonal, por ejemplo 12/10/94. Su longitud



siempre es de 8 caracteres.

Logical

Su contenido es una letra que representa un valor de verdadero, falso o de sí y no (T, F, Y y N, respectivamente).



Memo

Un campo memo puede contener texto de cualquier longitud (existiendo tan sólo la limitación del espacio disponible en el disco rígido). Se asigna una longitud fija a este campo (10 caracteres), y al capturar información se genera un archivo adicional cuya extensión es FPT donde se guardarán los datos.



Para editar el campo memo se le teclea Ctrl+W , lo mismo para grabar información

General

Este tipo de campo puede ser utilizado para guardar imágenes, sonido, documentos creados con un procesador de textos, hojas de cálculo, o cualquier otro tipo de datos.



Ejercicios:

Selecciona el tipo de datos de Visual Fox que puede emplearse para registrar los datos siguientes:

Con_Plano _____

Res_Presidencial_____

Edad:_____

Clave_concatenada:_____

Fecha_entrevista:_____

Grabación:_____

En_Sistema:_____

Tot_vivendas:_____

Clv_inventario:_____

Ejercicio: Crear una tabla con la estructura siguiente, capturar 2 registros

CAMPO	TIPO	LONGITUD	DESCRIPCION
C_MUN	C	3	Clave de municipio
P_TOT	N	6	Población total
F_REG	D	8	Fecha de registro
EN_SIS	L	1	Capturado en sistema
DIA_M	M		Diagnóstico Municipal
ESCUDO	G		Escudo del municipio (Se insertará el escudo del Mpio)

El nombre de la tabla será MUNICIPIOS.DBF

1.7 COPY STRUCTURE (Comando)

Crea una nueva tabla vacía pero con la misma estructura que la tabla fuente.

Sintaxis

COPY STRUCTURE TO Nombre_de_Tabla [FIELDS Lista_de_campos]

Parámetros

Nombre_de_Tabla

Especifica el nombre de la nueva tabla vacía que se desea crear.

FIELDS *Lista_de_campos*

Especifica que solamente se copien a la tabla nueva los campos cuyos nombres se especifican en *Lista_de_campos*. Si omite FIELDS *Lista_de_campos*, se copiarán todos los campos a la nueva tabla.

Ejercicios:

1. A partir de la tabla 2014.DBF copiar la estructura a la tabla E1_2014 :
2. Abrir la tabla creada
3. Repetir el ejercicio únicamente que copiando los 2 primeros campos

1.8 MODIFY STRUCTURE (Comando)

Muestra el [Diseñador de tablas](#), que permite modificar la estructura de una tabla.

MODIFY STRUCTURE

Observaciones

Si no hay ninguna tabla abierta en el área de trabajo seleccionada actualmente, se muestra el cuadro de diálogo Abrir, que permite elegir la tabla que se desea modificar.

Cambios que se pueden realizar a los campos



.EJERCICIOS

A partir de la tabla ME_2014.dbf realizar los siguientes cambios a la estructura:

1. Insertar las columnas de CONSECUTIVO Texto(3) Al principio de la estructura y OBSERVACIONES Texto(50) al final
2. Agregar los campos C1 Texto (1) , C3 Numérico (2) , C8 Fecha(8)
3. Eliminar los campos C3 y C8
4. Cambiar de nombre al campo denominado C1 por el nombre de Responsable
5. Reducir el tamaño del campo de nombre de la entidad a 40 caracteres
6. Cambiar el tipo del campo de Carácter a Numérico del campo Altitud
7. Ampliar el campo ámbito 3 caracteres

1.9 IMPORTAR Y EXPORTAR

La función Importar se utiliza para incorporar a una aplicación objetos, documentos u otro tipo de archivos que fueron creados en otra **aplicación, o que pertenecen a versiones antiguas** de la misma aplicación.

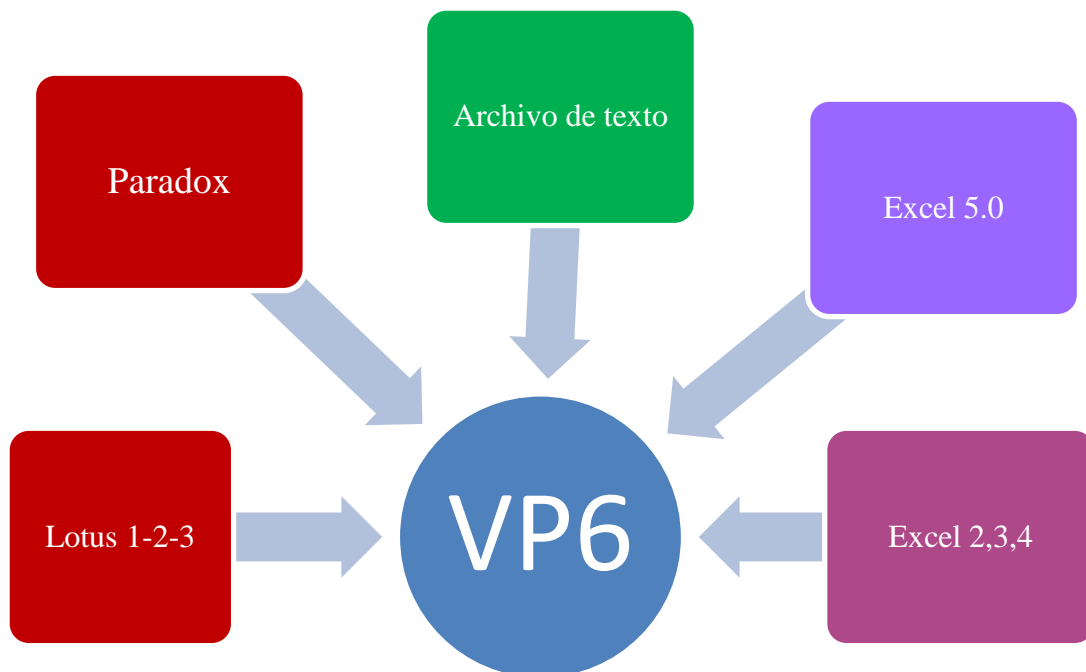
El pase puede ser directo o a través de archivos compatibles para ambos programas. El programa emisor exporta los datos y el programa receptor los importa.

Por ejemplo, supongamos que se utiliza el administrador de emails Microsoft Outlook. Allí disponemos de todos nuestros emails.

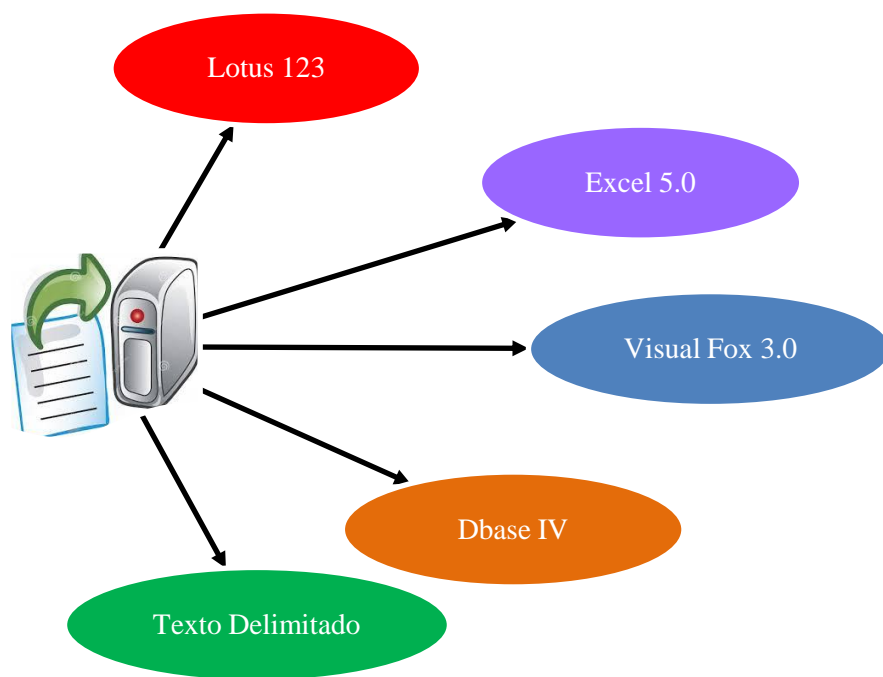
Si queremos cambiar de administrador de email, de Outlook hacia el Pegasus Mail, tendremos que encontrar la forma de transferir todos esos emails que se encuentran en "formato" de Outlook, hacia el Pegasus. Para ello existe la función Exportar en Outlook: uno exporta toda información de emails a una formato común a las dos aplicaciones, es decir, un formato que pueda interpretarse tanto por Outlook como por Pegasus.

De esta manera, con la función Importar de Pegasus, podremos llevar todos nuestros emails para administrarlos con este programa.

<http://www.alegsa.com.ar/Dic/importar.php>



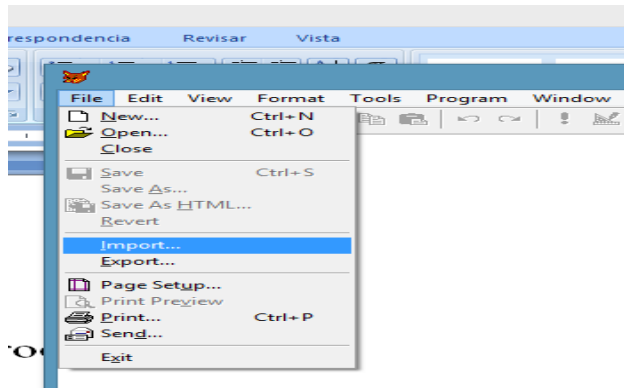
Importando a Fox desde diferentes formatos



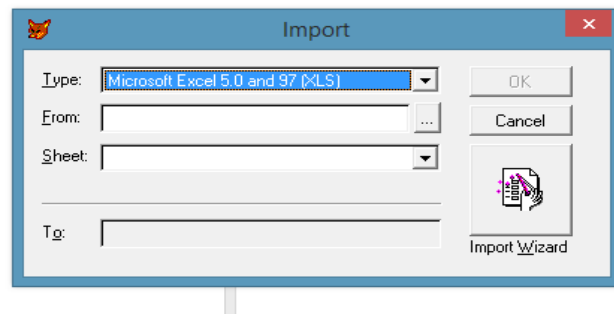
Exportando de Fox a otros formatos

1.8.1. Procedimiento para Importar: Archivo de texto a FOX

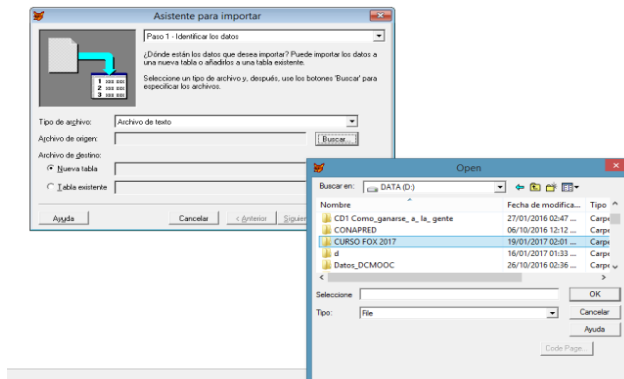
2. Seleccionar File > Import



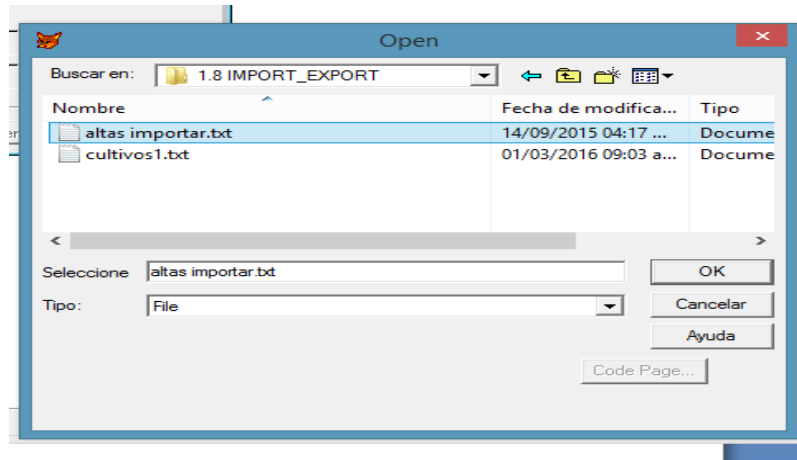
3. Dar clic en Import Wizard (es el asistente de importación)



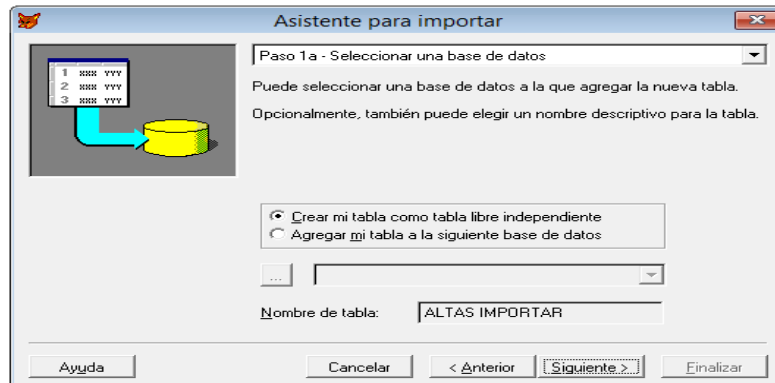
4. Seleccionar el tipo de archivo (en este caso es archivo de texto), a continuación dar clic en examinar e indicar la carpeta y el nombre del archivo



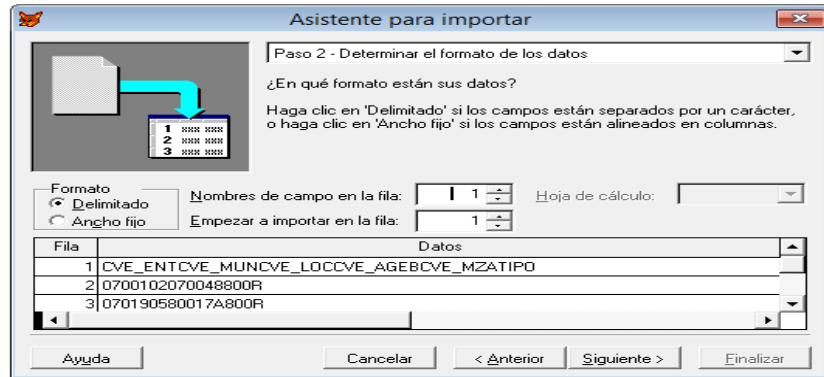
4. Seleccionar el nombre del archivo



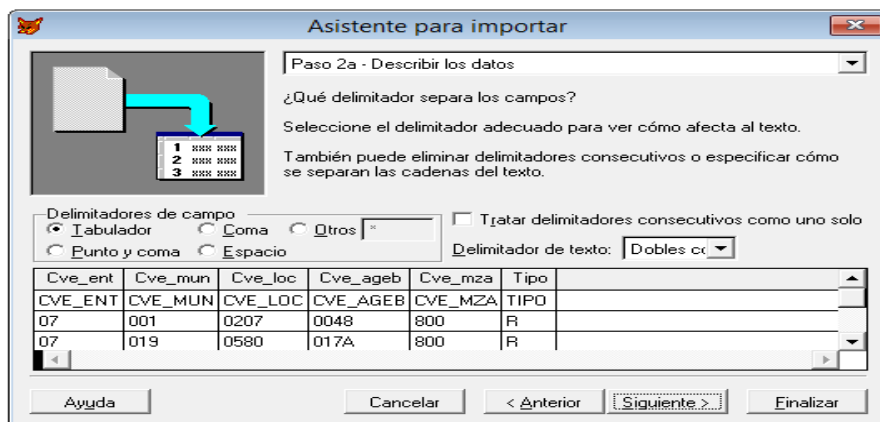
5. Darle siguiente hasta que aparezca esta pantalla



6. Escribirle 1 a la casilla de verificación más hacia arriba del cuadro de diálogo siguiente



7. Darle siguiente, siguiente... hasta que aparezca Finalizar activado



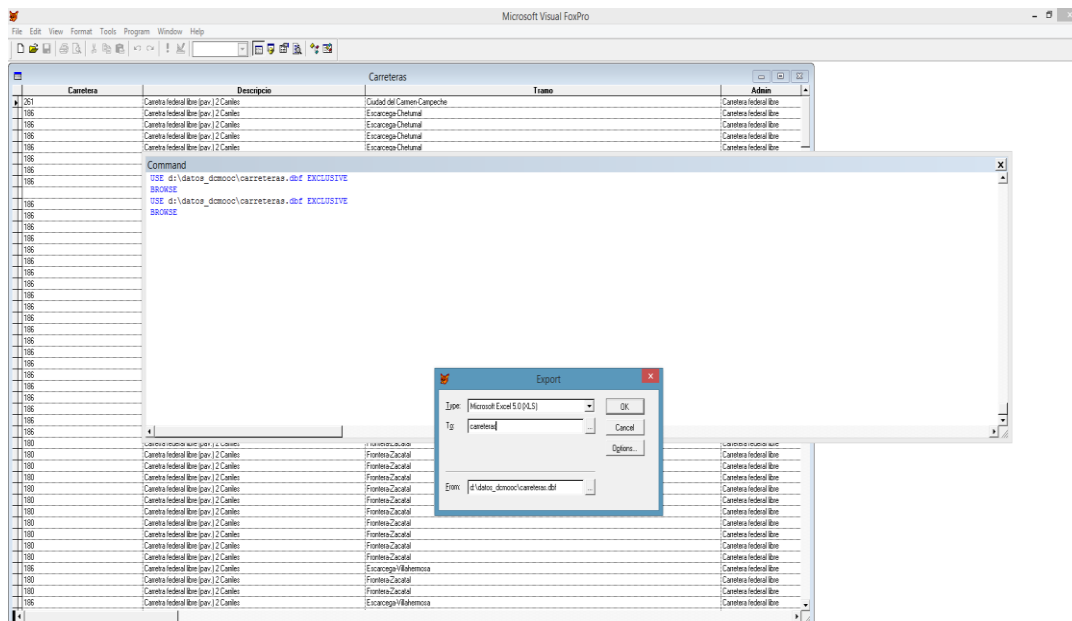
8. Una vez que se ha realizado el procedimiento procedemos a verificar la integridad de los datos, es decir que aparezcan todos los campos y los registros de manera correcta

Altas_importar						
Cve_ent	Cve_mun	Cve_loc	Cve_ageb	Cve_mza	Tipo	
CVE_ENT	CVE_MUN	CVE_LOC	CVE_AGE	CVE_MZA	TIPO	
07	001	0207	0048	800	R	
07	019	0580	017A	800	R	
07	027	0610	0044	800	R	
07	030	0141	0105	800	R	
07	031	0087	0206	800	R	
07	041	0563	0039	800	R	
07	052	0939	0263	800	R	
07	055	1336	0098	800	R	
07	075	0234	0060	800	R	
07	079	0018	006A	800	R	
07	083	0173	0039	800	R	
07	093	0081	0015	800	R	
07	094	0158	0047	800	R	
07	094	0166	0032	800	R	
07	094	0173	0047	800	R	
07	094	0174	0032	800	R	
07	096	0162	0037	800	R	
07	099	0601	0128	800	R	
07	105	0038	0023	800	R	
07	106	0054	0023	800	R	
07	109	0079	0037	800	R	

1.8.2. Exportando hacia Excel

Para realizar la exportación de una tabla, seguir estos pasos

1. Establecer la ruta por definición con SET DEFA
SET DEFAULT TO "d:\curso fox 2017\ejercicios\1.8 import_export"
2. Seleccionar File
3. Seleccionar Export
4. Seleccionar el tipo de documento que desea generarse, el más común es Microsoft Excel 5.0
5. Indicar correctamente la carpeta en donde será creado el archivo
6. Darle OK
7. Abrir el Excel para verificar si fue correctamente exportado



	A1	B	C	D	E	F
1	cve_ent	cve_mun	cve_loc	cve_ageb	cve_mza	tipo
2	CVE_ENT	CVE_MUN	CVE_LOC	CVE_AGEB	CVE_MZA	TIPO
3	07	001	0207	0048	800	R
4	07	019	0580	017A	800	R
5	07	027	0610	0044	800	R
6	07	030	0141	0105	800	R
7	07	031	0087	0206	800	R
8	07	041	0569	0039	800	R
9	07	052	0939	0263	800	R
10	07	065	1336	0098	800	R
11	07	075	0234	0060	800	R
12	07	079	0018	006A	800	R
13	07	083	0173	0039	800	R
14	07	093	0081	0016	800	R
15	07	094	0158	0047	800	R
16	07	094	0166	0032	800	R
17	07	094	0173	0047	800	R
18	07	094	0174	0032	800	R
19	07	096	0162	0037	800	R
20	07	099	0601	0128	800	R
21	07	105	0038	0023	800	R
22	07	105	0054	0023	800	R
23	07	109	0079	0037	800	R
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						

Comprobación de exportación correcta

Ejercicios:

1. Importar el archivo de cultivos1.txt a DBF
2. Exportar a formato de texto delimitado el archivo generado

UNIDAD II

Visualizar registros

INTRODUCCIÓN A LA

UNIDAD II

El primer comando que se revisará es el BROWSE es de uso muy frecuente para el usuario de FOX

El segundo tema es con respecto al alcance o SCOPE, nos permitirá conocer cómo debemos indicar a cuántos registros afectará el comando que indiquemos: ALL, NEXT n, n, REST

El equivalente al tema de filtros en Excel es el SET FILTER.

Otro tema a revisar son comandos o funciones, con respecto a la ubicación de registros.

Por último, se revisará la función DELETED (), para determinar si un registro está marcado para borrar

2.1. BROWSE (Comando)

Browse significa hojear o dar un vistazo. Muestra los registros de la tabla en uso o seleccionada.

```
BROWSE [FIELDS Lista_de_campos] [FONT cNombre_de_fuente [,  
nTamaño_de_fuente]]  
[FOR lExpresión1]
```

Parámetros

FIELDS *Lista_de_campos*

Especifica los campos que aparecerán en la ventana Examinar. Los campos se muestran en el orden especificado en *Lista_de_campos*..

Si omite FIELDS, se mostrarán todos los campos de la tabla en el orden en que aparecen en la estructura de la tabla.

FONT *cNombre_de_fuente* [, *nTamaño_de_fuente*]

Especifica la fuente y el tamaño de fuente de la ventana Examinar. La expresión de caracteres *cNombre_de_fuente* es el nombre de la fuente y la expresión numérica *nTamaño_de_fuente* es el tamaño de la fuente. Por ejemplo, la cláusula siguiente especifica la fuente Courier de 16 puntos para los campos que se muestran en la ventana Examinar:

```
FONT 'Courier',16
```

Si incluye la cláusula FONT pero omite el tamaño de fuente *nTamaño_de_fuente*, se usará una fuente de 10 puntos en la ventana Examinar. Si omite la cláusula FONT, se usará MS Sans Serif de 8 puntos.

Si la fuente especificada no está disponible, se sustituirá por una fuente de características similares.

FOR *lExpresiónl*

Especifica una condición por la cual sólo se mostrarán en la ventana Examinar aquellos registros para los que *lExpresiónl* sea verdadera.

.

Incluya FOR si desea mover el puntero de registro hasta el primer registro que cumpla la condición. Incluya REST si desea dejar el puntero de registro en su posición actual.

EVALUACIÓN

1.-Muestra todos los campos de la tabla en uso:_____

2.-Evita la edición de un campo:_____

3.-Seleccionamos los registros que cumplan una determinada condición:_____

4.-Tipo de dato que debe ir en el tamaño de la fuente:_____

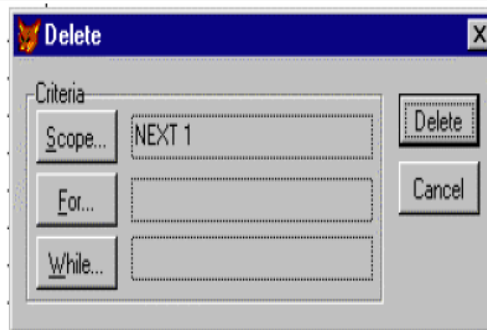
EJERCICIOS

1. Abrir el archivo de Cigel ubicado en D:\CURSO FOX 17\II
2. Mostrar todos los registros
3. Mostrar los campos cve_mun, cve_loc, nom_loc
4. Mostrar los campos cve_mun, cve_loc con fuente Arial, tamaño 16
5. Mostrar los campos cve_mun, cve_loc, nom_loc para el municipio “008”
6. Mostrar los campos cve_mun, cve_loc, nom_loc para el municipio “003” o el municipio “024”
7. Mostrar los campos cve_mun, cve_loc y los campos que tiene latitud, longitud y altitud para municipios diferentes a “003”

2.1.1.ALCANCE (ALCANCE)



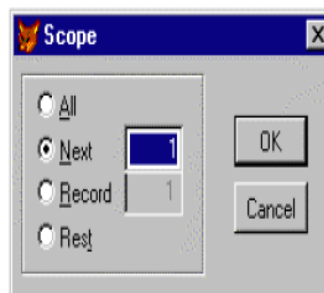
El alcance se refiere a la cantidad de registros que serán afectados por el comando que el usuario esté utilizando



Nota: Si hace clic en el botón Delete borrará el registro en donde se encuentre el cursor.

- Para definir el criterio que se usará para borrar los registros comience haciendo un clic sobre el botón Scope (Alcance).

El alcance se refiere a la cantidad de registros sobre los cuales tendrá efecto el comando. Los diferentes alcances son:



All: El comando trabajará sobre todos los registros de la tabla, independientemente de la posición del cursor.

Next: El comando trabajará en los siguientes N registros, donde N se especifica en el cuadro de texto adjunto a la opción. Considere que la numeración comienza a partir del registro actual, por lo que Next 1 será precisamente el registro donde se encuentra el cursor.

Record: El comando trabajará exclusivamente en el registro que se encuentre colocado en la posición especificada en el cuadro de texto adjunto. Tiene la desventaja de que se debe conocer exactamente la posición que ocupa el registro a borrar. En todas las versiones de FoxPro se puede acceder a un registro a través del número que ocupa en la tabla. Este es un número consecutivo que se asigna al registro a medida que se van agregando datos a la tabla. La marca de borrado no altera esta numeración por sí sola.

Rest: El comando actuará a partir del registro actual y hasta el final del archivo.

Esquema de las diferentes formas de alcance

RECNO()	ALL	NEXT	RECORD	REST
1	1			
2	2		2	
3	3	1		
4	4	2		1
5	5	3		2
6	6		6	3
7	7			4

2.2. SET FILTER (Comando)

Especifica una condición que deben cumplir los registros de la tabla actual para estar accesibles.

SET FILTER TO [lExpresión]

Parámetros

lExpresión

Especifica la condición que deben cumplir los registros.

Observaciones

Una vez ejecutado SET FILTER, solamente están disponibles en la tabla los registros que satisfacen la condición especificada con la expresión lógica lExpresión.

Ejercicios

- 1. Abrir el archivo de CIGEL de la unidad II**
- 2. Filtrar los registros correspondientes a la carta D15B11**
- 3. Filtrar las localidades que comiencen con la cadena “San”**

2.3.LOCATE (Comando)

Busca secuencialmente en la tabla el primer registro que coincida con la expresión lógica especificada. Se utiliza junto con el comando BROWSE para visualizar

Sintaxis

LOCATE [FOR lExpresión1] [Alcance]

Parámetros

FOR lExpresión1

LOCATE busca secuencialmente en la tabla actual el primer registro que coincida con la expresión lógica lExpresión1.

Alcance

Especifica el intervalo de objetos que se buscará. Solamente se buscarán los registros incluidos en el intervalo. Las cláusulas de alcance son: ALL, NEXT nRecords, RECORD nRecordNumber y REST. Los comandos que incluyen Alcance sólo actúan sobre la tabla del área de trabajo activa.

El alcance predeterminado de LOCATE es ALL (todos los registros).

Ejercicios

1. Ubicar en la tabla el registro correspondiente al municipio 032 y localidad 0091
2. Visualizar el registro
3. Mostrar el primer registro que tenga como valor del campo longitud 0924057

2.4.COUNT (Comando)

Cuenta los registros de una tabla.

Sintaxis

COUNT [Alcance] [FOR *lExpresión1*] [TO *Nombre_de_variable*]

Parámetros

Alcance

Especifica el intervalo de registros que se va a incluir en la cuenta. Las cláusulas de alcance son: ALL, NEXT *nRecords*, RECORD *nRecordNumber* y REST.

El alcance predeterminado de COUNT es ALL (todos los registros).

FOR *lExpresión1*

Especifica que solamente se cuentan los registros que satisfacen la condición lógica *lExpresión1*. Incluir FOR le permite contar registros condicionalmente y desechar expresión lógica *lExpresión2* dé como resultado verdadero (.T.).

TO *Nombre_de_variable*

Especifica la variable o la matriz en que se almacena la cuenta de registros. Si la variable especificada no existe, Visual FoxPro la creará.

Observaciones

COUNT cuenta los registros que estén en el alcance de registros para los que se cumplen las condiciones FOR o WHILE. Si SET TALK está establecido en ON, se mostrará la cuenta de registros.

Si SET DELETE está establecido en OFF, los registros marcados para su eliminación se incluirán en la cuenta.

Ejercicios

1. Contar el número de registros que tienen plano
2. Contar el número de registros del municipio 003 y almacenarlo en la variable num_reg003

2.5. GO | GOTO (Comando)

Coloca el puntero de registro en el número de registro especificado de una tabla.

```
GO [RECORD] nRecordNumber
```

–O bien–

```
GO TOP | BOTTOM
```

–O bien–

```
GOTO [RECORD] nRecordNumber
```

–O bien–

```
GOTO TOP | BOTTOM
```

Parámetros

RECORD *nRecordNumber*

Especifica el número de registro físico al que desea mover el puntero de registro. Puede omitir por completo GO o GOTO y especificar simplemente el número de registro. Si especifica solamente el número de registro, podrá mover el puntero solamente en el área de trabajo actual.

TOP

Sitúa el puntero de registro en el primer registro de la tabla.

BOTTOM

Sitúa el puntero de registro en el último registro de la tabla.

Observaciones

GO y GOTO pueden utilizarse indistintamente. Estos comandos actúan sobre la tabla del área de trabajo actual a menos que especifique otra área de trabajo con la cláusula IN.

2.6.SKIP

Permite brincar al siguiente registro. n es el incremento o decremento, según se especifique

Sintaxis

[SKIP n]

Argumentos

n

Número de elementos que se han de "Saltar"

2.7. RECNO() (Función)

Devuelve el número del registro actual de la tabla actual o la especificada.

RECNO()

Valores devueltos

Numeric

Observaciones

El registro actual es el registro en el cual se encuentra el puntero de registro.

2.8 DELETED() (Función)

Devuelve un valor lógico que indica si el registro actual está marcado para su eliminación.

DELETED()

Observaciones

Si el registro está marcado para eliminación, DELETED() devolverá el valor verdadero (.T.); de lo contrario, DELETED() devolverá el valor falso (.F.).

UNIDAD III

Modificar registros

INTRODUCCIÓN A LA

UNIDAD III

En esta unidad se verán los comandos necesarios para hacer la modificación de registros:

Actualización ,empleando el comando REPLACE

Eliminación de registros: los comandos empleados pueden ser :
DELETE, PACK, ZIP, RECALL

Inserción de registros en blanco de manera intermedia (INSERT BLANK) o al final del archivo (APPEND BLANK)

Un comando importante para agregar registros procedentes de una tabla fuente a una destino con APPEND FROM

Por último se verá el procedimiento para ordenar registros, ya sea por medio de uno o varios criterios, de forma ascendente o descendente

3.1.REPLACE (Comando)



Actualiza los registros de una tabla. Este comando modifica el valor del campo sobre el cual el usuario especifica, por lo que debemos tener mucho cuidado de no manejar el archivo fuente, sino una copia

REPLACE [Alcance] Nombre_de_campo1 WITH eExpresión1 [FOR IExpresión1]

Parámetros

Alcance

Especifica un rango de registros que se va a sustituir. Solamente se sustituyen los registros incluidos dentro del rango. Las cláusulas de alcance son: ALL, NEXT *nRecords*, RECORD *nRecordNumber* y REST.

El alcance predeterminado de REPLACE es el registro actual (NEXT 1).

Nombre_de_campo1 WITH eExpresión1

Especifica que los datos de *Nombre_de_campo1* se sustituyan por el valor de la expresión *eExpresión1*; que los datos de *Nombre_de_campo2* se sustituyan por el valor de la expresión *eExpresión2*; y así sucesivamente.

FOR *lExpresión1*

Especifica que los campos designados se sustituyan únicamente en los registros para los que *lExpresión1* se evalúe como verdadero (.T.). La inclusión de FOR le permite sustituir registros de forma condicional, descartando los registros no deseados.

Observaciones

REPLACE sustituye los datos de un campo con el valor de una expresión.

Ejercicios

1. Agregarle un cero antes a la clave de ageb
2. Reemplazar la clave concatenada, implica la modificación de la estructura, de manera previa

3.2 DELETE (Comando)

Marca los registros que se van a eliminar. Los registros marcados para su eliminación no se eliminarán físicamente de la tabla hasta que se ejecute PACK. Los registros marcados para su eliminación pueden recuperarse (o quitar las marcas) con RECALL

Sintaxis

DELETE [Alcance] [FOR lExpresión1]

Parámetros

Alcance

Especifica un intervalo de registros que desea marcar para su eliminación. Las cláusulas de alcance son: ALL, NEXT *nRecords*, RECORD *nRecordNumber* y REST.

El alcance predeterminado para el comando DELETE es el registro actual (NEXT 1).

FOR lExpresión1

Especifica una condición por la cual solamente se marcan para su eliminación los registros que satisfacen la condición lógica *lExpresión1*.

Ejemplo

El ejemplo siguiente abre la tabla customer de la base de datos testdata. Se utiliza DELETE para marcar para eliminación todos los registros en los que el campo country contenga USA. Se muestran todos los registros marcados para ser eliminados. Se utiliza RECALL ALL para quitar las marcas de todos los registros marcados para eliminar.

CLOSE DATABASES

OPEN DATABASE (HOME(2) + 'Data\testdata')

USE customer && Opens Customer table

DELETE FOR country = 'USA' && Mark for deletion

CLEAR

BROW FIELDS company, country FOR DELETED() && List marked records

RECALL ALL && Unmark all records marked for deletion

3.3. PACK (Comando)

Borra definitivamente todos los registros marcados para eliminar de la tabla actual y reduce el tamaño de un archivo memo asociado a la tabla.

Sintaxis

PACK

Parámetros

Sin parámetros

Observaciones

Cuando se ejecuta PACK, Microsoft Visual FoxPro copia todos los registros que no están marcados para borrar a una tabla temporal. Cuando se termine de ejecutar PACK, Visual FoxPro eliminará la tabla original del disco y cambiará el nombre de la tabla temporal por el nombre de la tabla original. PACK necesita utilizar la tabla de forma exclusiva. Para obtener más información acerca de cómo abrir una tabla en red de forma exclusiva, vea SET EXCLUSIVE

Precaución Tenga cuidado de marcar solamente los registros que ya no va a necesitar. No hay ninguna forma de recuperar los registros eliminados después de utilizar PACK.

3.4. ZAP (Comando)

Elimina todos los registros de la tabla actual y deja sólo la estructura de la tabla.

Sintaxis

ZAP

Observaciones

El uso de ZAP equivale a especificar DELETE ALL y a continuación PACK, pero ZAP es mucho más rápido.

Si SET SAFETY está establecido en ON, Microsoft Visual FoxPro le preguntará si desea eliminar los registros de la tabla actual.

Precaución Los registros de la tabla actual eliminados con ZAP no pueden recuperarse.

3.5 RECALL

Quita la marca de los registros marcados para eliminación en la tabla seleccionada.

RECALL [Alcance] [FOR lExpresión1]

Parámetros

Alcance

Especifica un intervalo de registros para recuperar. Solamente se recuperan los registros incluidos en el intervalo especificado. Las cláusulas de alcance son: ALL, NEXT nRecords, RECORD nRecordNumber y REST.

El alcance predeterminado de RECALL es el registro actual (NEXT 1).

FOR lExpresión1

Especifica que sólo se recuperarán los registros para los que el valor de lExpresión1 sea verdadero (.T.). Esta opción permite filtrar los registros no deseados.

Observaciones

RECALL puede utilizarse para recuperar registros, siempre y cuando no se haya ejecutado PACK o ZAP.

Precaución Una vez empaquetado el archivo, todos los registros marcados para eliminación se pierden para siempre.

Ejemplo

El ejemplo siguiente abre la tabla customer de la base de datos testdata. DELETE – SQL se utiliza para marcar todos los registros donde el campo country contenga USA para su eliminación. Se muestran todos los registros marcados para ser eliminados. Se usa RECALL ALL para quitar las marcas de todos los registros marcados para eliminar

Ejemplo de uso del RECALL

CLOSE DATABASES

OPEN DATABASE (HOME(2) + 'data\testdata')

USE customer && Opens Customer table

DELETE FROM customer WHERE country = 'USA' && Mark for deletion

CLEAR

3.6 INSERT BLANK

Nos permite insertar un registro en blanco, entre un registro y otro

Sintaxis

Observaciones

Es muy útil cuando se combina su uso con set carry on, para que duplique el contenido del registro de donde se halle el puntero

BROW FIELDS company, country **FOR DELETED() &&** List marked records

RECALL ALL && Unmark all records marked for deletion

3.7 APPEND BLANK

Cuando crea por primera vez una tabla de Visual FoxPro, está abierto pero vacío. Si intenta almacenar datos en una tabla sin crear primero un registro en la tabla, no sucede nada. El primer paso para agregar registros a una nueva tabla es agregar filas para almacenar los nuevos datos.

También puede usar el comando APPEND BLANK seguido del comando REPLACE para agregar un registro en blanco a una tabla y luego almacenar datos en un campo. APPEND BLANK agrega un nuevo registro en blanco a una tabla. El comando REPLACE reemplaza el valor actual de un campo, incluso un campo vacío, con un nuevo valor.

El comando REPLACE requiere:

- Una mesa abierta.
- Un registro existente.
- El nombre del campo en el que se almacenará el valor.
- Un valor para cada campo que es válido para el tipo de datos del campo.

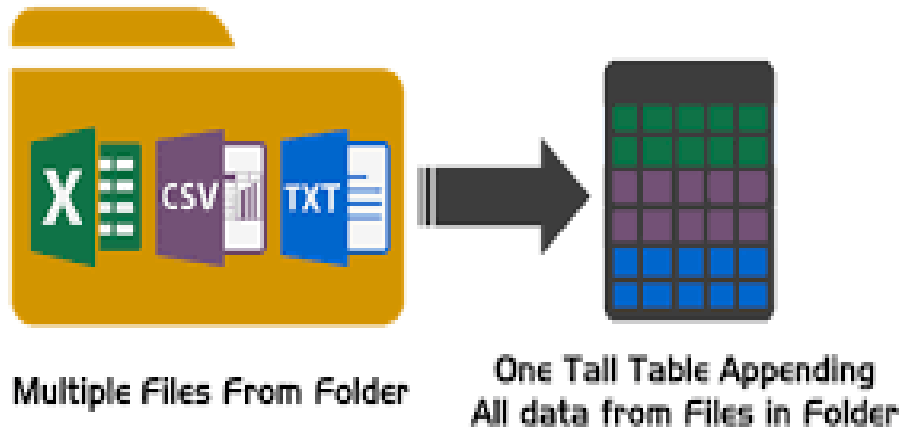
El siguiente ejemplo utiliza el comando APPEND para crear un registro en el que puede almacenar datos mediante el comando REPLACE:

APPEND BLANK && registro ahora disponible

REPLACE lastname WITH "SMITH" & & almacenar el valor del carácter en el campo

3.8.APPEND FROM (Comando)

Agrega registros, desde otro archivo, al final de la tabla seleccionada actualmente.



Ejemplo gráfico que muestra lo que hace Append. Tenemos múltiples archivos en un folder: Archivo de Excel, Archivos en formato CSV, Archivos en formato de texto. Se hacen APPENDs sucesivos y se conforma una nueva tabla con los elementos anteriores.

Para emplear este comando debemos de tener los archivos en una sola carpeta para facilitar el proceso.

Sintaxis

APPEND FROM FileName | ? [FIELDS Lista_de_campos] [FOR lExpresión]

Parámetros

FileName

Especifica el nombre del archivo desde el que desea agregar registros. Si no incluye ninguna extensión de archivo, se asumirá una tabla de Visual FoxPro y la extensión predeterminada .dbf. Si desea agregar desde una tabla de Visual FoxPro, se agregarán los registros de la tabla que estén marcados para su eliminación, independientemente de la configuración de SET DELETED.

?

Muestra el cuadro de diálogo Abrir, en el que puede elegir una tabla desde la que agregar registros.

FIELDS Lista_de_campos

Especifica a qué campos se van a agregar datos.

FOR lExpresión

Agrega un nuevo registro por cada registro de la tabla actualmente seleccionada para la que *lExpresión* dé como resultado verdadero (.T.). Los registros se agregan hasta alcanzar el final de la tabla seleccionada. Si omite FOR, todo el archivo de origen se agrega a la tabla seleccionada actualmente.

Ejercicios:

- 1. Crear las tablas con sus registros correspondientes como se muestra en el esquema**

UNO.DBF

CVE
001
002

DOS.DBF

CVE
003
004

TRES.DBF

CVE
001
002
003
004

3.9 SORT

Ordena los registros de la tabla seleccionada actualmente y envía los registros ordenados a una tabla nueva.

Sintaxis

`SORT TO Nombre_de_Tabla ON Nombre_de_campo1 [/A | /D] [/C]`

`[, Nombre_de_campo2 [/A | /D] [/C] ...] [ASCENDING | DESCENDING]`

`[Alcance] [FOR lExpresión1]`

`[FIELDS Nombre_de_campoList]`

Parámetros

`Nombre_de_Tabla`

Especifica el nombre de la nueva tabla que contiene los registros ordenados. La extensión .dbf se asigna automáticamente si el nombre de archivo incluido no tiene ninguna extensión.

`ON Nombre_de_campo1`

Especifica el campo de la tabla seleccionada actualmente en el que se basa el orden. El contenido y el tipo de datos del campo determinarán el orden de los registros de la tabla. De forma predeterminada, la ordenación se hará en sentido ascendente. No podrá ordenar por campos de tipo Memo o General.

Puede incluir nombres de campo adicionales (Nombre_de_campo2, Nombre_de_campo3) para ordenar aun más la tabla nueva. El primer campo Nombre_de_campo1 es el campo de ordenación principal, el segundo campo Nombre_de_campo2 es el campo de ordenación secundario, etc.

[/A | /D]

Puede especificar un sentido ascendente o descendente para cada uno de los campos que incluya en la ordenación. La opción /A especifica orden ascendente. La opción /D especifica orden descendente. /A o /D pueden incluirse con cualquier tipo de campo.

En el ejemplo siguiente se crea una tabla nueva denominada clients. La tabla orders se ordena según el campo order_date en orden ascendente y según el campo freight en orden descendente

```
USE orders
```

```
SORT TO clients ON order_date/A,freight/D
```

ASCENDING

Especifica un orden ascendente para todos los campos que no vayan seguidos de la opción /A.

DESCENDING

Especifica un orden descendente para todos los campos que no vayan seguidos de la opción /D.

Si no incluye ASCENDING o DESCENDING, asumirá el sentido de ordenación predeterminado, que es ascendente.

Alcance

Especifica un intervalo de registros que desea ordenar. Las cláusulas de alcance son: ALL, NEXT nRecords, RECORD nRecordNumber y REST.

El alcance predeterminado de SORT es ALL (todos los registros).

FOR lExpresión1

Especifica que solamente se incluyan en la ordenación los registros de la tabla actual para los que la condición lógica lExpresión1 da como resultado verdadero (.T.). Incluir FOR le permite ordenar registros condicionalmente, filtrando los registros no deseados.

FIELDS Nombre_de_campoList

Especifica los campos procedentes de la tabla original que se van a incluir en la nueva tabla creada por SORT. Si no se incluye la cláusula FIELDS, se incluirán todos los campos de la tabla original en la nueva tabla.

Precaución Asegúrese de que tiene suficiente espacio en disco para la nueva tabla y para los archivos temporales creados al ordenar. El espacio en disco necesario para realizar una ordenación puede ser, como máximo, tres veces el tamaño de la tabla de origen. La cantidad de espacio disponible en disco puede determinarse con DISKSPACE() y SYS(2020). Si agota el espacio del disco durante una ordenación, Visual FoxPro mostrará un mensaje de error y se eliminarán los archivos temporales.

Es posible que los campos de tipo Character que contienen números y espacios no se ordenen en el sentido esperado. Los campos numéricos se llenan de derecha a izquierda, con espacios vacíos a la izquierda. Por el contrario, los campos alfanuméricos se llenan de izquierda a derecha, con espacios vacíos a la derecha.

Por ejemplo, si dos registros de una tabla contienen un campo de tipo Character con el valor 1724 en un registro y el valor 18 en otro, y se ordena la tabla en orden ascendente en este campo, se mostrará el registro que contiene 1724 antes que el registro que contiene 18. Esto se debe a que Visual FoxPro lee cada carácter de los campos de caracteres de izquierda a derecha; como 17 (de 1724) es menor que 18 (de 18), Visual FoxPro coloca 1724 en primer lugar. Para evitar este problema, siempre debe agregar ceros al principio de los números más bajos (0018) o convertir el campo en un campo de tipo Numeric.

UNIDAD IV

Funciones y comandos de configuración

INTRODUCCIÓN A LA

UNIDAD IV

En esta Unidad se revisarán la forma habitual para la evaluación de expresiones ?

Se analizarán las funciones de tipo matemático: la sumatoria SUM() o conteo de registros COUNT

Veremos también las funciones que nos permiten manipular cadenas: concatenar cadenas , usando el operador +, extraer una subcadena de una cadena con SUBSTR, el procedimiento para calcular la longitud de una cadena

Cambiar de minúsculas a mayúsculas y viceversa.

El cambio de tipo de datos es útil cuando se pretende reemplazar datos a campos en los que difiere el tipo , por ello se verán los comandos VAL () y STR ()

Por último, se revisarán los comandos de configuración más comunes

4.1. Comando ?

Este comando es de vital importancia para la evaluación de las expresiones, verificar los ejemplos de operadores en el ANEXO

4.2.1. Comando SUM

Totaliza los valores de todos los campos numéricos especificados en la tabla actual seleccionada.

Sintaxis

```
SUM [eExpresiónList] [Alcance] [FOR lExpresión1] [TO  
MemNombre_de_variableList]
```

Parámetros

eExpresiónList

Especifica uno o varios campos o expresiones de campo que se van a totalizar. Si se omite la lista de expresiones de campo, se totalizarán todos los campos numéricos.

Alcance

Especifica un intervalo de registros que se van a utilizar en el total. Las cláusulas de alcance son: ALL, NEXT *nRecords*, RECORD *nRecordNumber* y REST.

El alcance predeterminado de SUM es ALL (todos los registros).

FOR lExpresión1

Especifica que solamente se incluyen en el total los registros para los cuales la condición lógica *Expresión1* da como resultado verdadero (.T.). La inclusión de FOR le permite calcular totales de registros de forma condicional, descartando los registros no deseados.

TO MemNombre_de_variableList

Almacena cada total en una variable. Si especifica en *MemNombre_de_variableList* el nombre de una variable que no existe, Visual FoxPro la creará automáticamente. Separe los nombres de variables en la lista con comas.

Ejemplo

El ejemplo siguiente muestra los totales de los campos `in_stock` y `on_order` de la tabla `products`, junto con una suma de los dos totales.

```
CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'data\testdata')
USE products  && Opens Products table

SUM in_stock, on_order, in_stock+on_order ;
    TO gnInStock, gnOnOrder, gnUnits

CLEAR
?
? 'Total in stock : ', gnInStock  && Displays 3119.00
? 'Total on order:  ', gnOnOrder  && Displays 780.00
? 'Total # units :  ', gnUnits    && Displays 3899.00
```

4.2.2. Comando AVERAGE

Calcula el promedio aritmético de expresiones numéricas o campos.

AVERAGE [ExpresiónList] [Alcance] [FOR IExpresión1] [TO VarList]

Especifica las expresiones a promediar. ExpresiónList puede ser una lista de campos de la tabla separados por comas, o expresiones numéricas que implican campos de la tabla.

Alcance

Especifica el registro o rango de registros a incluir en el promedio. Sólo se promedian los registros que están dentro del rango de registros especificados por el ámbito. Las cláusulas de ámbito son: ALL, NEXT nRecords, RECORD nRecordNumber y REST. El ámbito predeterminado para AVERAGE es TODOS los registros.

Los comandos que incluyen Alcance funcionan sólo en la tabla del área de trabajo activa.

FOR IExpresión1

Especifica una condición por la cual sólo se incluyen los registros que satisfacen la condición lógica IExpresión. Este argumento le permite filtrar registros no deseados.

TO VarList

Especifica la lista de variables o elementos de matriz a los que se almacenan los resultados del promedio.

4.2.3. VAL()

Devuelve un valor numérico a partir de una expresión de caracteres compuesta por números.

VAL(*cExpresión*)

Parámetros

cExpresión

Especifica una expresión de caracteres compuesta por un máximo de 16 números. Si en *cExpresión* se incluyen más de 16 números, se redondeará.

Observaciones

Ejemplo

CLEAR

STORE '12' TO A

STORE '13' TO B

? VAL(A) + VAL(B) && Displays 25.00

STORE '1.25E3' TO C

? 2 * VAL(C) && Displays 2500.00

4.3.1 + Concatenar cadenas

La concatenación de las cadenas se puede hacer empleando el operador +, como se muestra en este ejemplo

```
campo1=" jose "  
campo2=" lopez "
```

```
resultado=alltrim(campo1)+" "+alltrim(campo2)
```

La función alltrim nos sirve para eliminar todos los espacios del campo1 y campo2

4.3.2. SUBSTR()

Devuelve una subcadena especificada por el usuario

SUBSTR(*cExpresión*, *nPosiciónInicial* [, *nCharactersReturned*])

Parámetros

cExpresión

Especifica la expresión de caracteres desde los que se devuelve la cadena de caracteres.

nPosiciónInicial

Especifica la posición en la expresión de caracteres. El primer carácter de *cExpresión* es la posición 1.

nCharactersReturned

Especifica el número de caracteres que se devuelve de *cExpresión*. Si se omite *nCharactersReturned*, los caracteres se devolverán hasta llegar al final de la expresión de caracteres.

Observaciones

SUBSTR() devuelve una cadena de caracteres de una expresión de tipo Character , a partir de la posición especificada en la expresión Character y hasta el número especificado de caracteres.

Ejemplo

```
STORE 'abcdefghijklm' TO mystring
```

```
CLEAR
```

```
? SUBSTR(mystring, 1, 5) && Displays abcde
```

```
? SUBSTR(mystring, 6) && Displays fghijklm
```


4.3.3. LEN()

Devuelve el número de caracteres de una expresión de caracteres.

LEN(cExpresión)

Parámetros

cExpresión

Especifica la expresión de caracteres cuyo número de caracteres devuelve LEN().

Observaciones

Utilice LEN() para determinar la longitud de una expresión de caracteres.

Ejemplo

Se abre la tabla customer de la base de datos testdata. Se usa LEN() para mostrar los anchos de los campos cust_id y contact.

CLOSE DATABASES

OPEN DATABASE (HOME(2) + 'Data\testdata')

USE Customer && Open customer table

CLEAR

? 'Width of contact field: '

?? LEN(contact)

? 'Width of cust_id field: '

? LEN(cust_id)

4.3.4 LIKE() (Función)

Determina si una expresión de caracteres coincide con otra expresión de caracteres.

LIKE(cExpresión Like, cCampo cadena)

Expresión1 puede contener caracteres comodín, tales como * y ?. El signo de interrogación (?) coincide con cualquier carácter único de cExpresión2 y el asterisco (*) coincide con cualquier número de caracteres. Puede mezclar y combinar cualquier número de caracteres comodín que desee en cExpresión Like

Campo cadena

Especifica la expresión de caracteres que LIKE() compara

Ejemplo

En el ejemplo siguiente se muestran todos los nombres de productos de la tabla products cuyas dos primeras letras sean "Ch".

CLOSE DATABASES

OPEN DATABASE (HOME(2) + 'Data\testdata')

USE products && Open Products table

CLEAR

? 'All product names with first two letters Ch:'

?

BROW FOR LIKE('Ch*', prod_name)

? prod_name

USE

4.3.5 LOWER() (Función)

Devuelve en letras minúsculas una expresión de caracteres especificada.

LOWER(cExpresión)

Parámetros

cExpresión

Especifica la expresión de caracteres convertida por LOWER().

Observaciones

LOWER() convierte todas las letras mayúsculas (A – Z) de la expresión a letras minúsculas (a – z). Todos los demás caracteres de la expresión permanecen sin cambios.

Ejemplo

```
STORE 'FOX' TO gcName
```

```
CLEAR
```

```
? LOWER(gcName) && Displays fox
```

4.3.6 UPPER() (Función)

Devuelve la expresión de caracteres especificada en mayúsculas.

UPPER(cExpresión)

Parámetros

cExpresión

Especifica la expresión de caracteres que UPPER() convierte a mayúsculas.

Observaciones

Todas las letras minúsculas (a – z) de la expresión de caracteres se convierten en mayúsculas (A – Z) en la cadena devuelta. Esto también es válido para caracteres ASCII altos, aunque la fuente actual no los muestre como caracteres acentuados en minúsculas.

Ejemplo

CLEAR

? UPPER('abcdefgh') && Displays ABCDEFGH

4.3.7 STR() (Función)

Devuelve el carácter equivalente a una expresión numérica especificada.

STR (nExpresión)

Valores devueltos

Character

Parámetros

nExpresión

Especifica la expresión numérica evaluada por STR() convertida en numérica

4.4.1.SET STATUS

Muestra o quita la barra de estado

SET STATUS ON | OFF

Parámetros

ON

Si SET STATUS está establecido a ON, se muestra la barra de estado basada en caracteres con el nombre del programa actualmente en ejecución (si lo hay), el alias de la tabla activa, la posición del puntero en el registro actual, el número de registros de la tabla y el estado de las teclas Insert, Bloq Num y Bloq Mayús. El estado de bloqueo del registro o del archivo también se muestra en la barra de estado cuando se abre la tabla para uso compartido. La barra de estado se actualiza cada vez que ejecuta un comando que cambia la información de estado.

OFF

(Predeterminado) Ejecute SET STATUS OFF para quitar la barra de estado.



|<c:>|A (c:\users|Record Unlocked|Rec: 20/399 | |NUMCAPS

4.4.2 SET CARRY



Determina si Visual FoxPro acarreará o no datos hacia adelante desde el registro actual al nuevo registro creado con INSERT, APPEND y BROWSE.

SET CARRY ON | OFF

-or-

SET CARRY TO [Lista_de_campos [ADDITIVE]]

Parámetros

ON

Acarrea datos de todos los campos de todas las áreas de trabajo hacia adelante desde el registro actual hasta un nuevo registro.

OFF

(Predeterminado) Impide que se transfieran datos de todos los campos al registro nuevo.

4.4.3 SET DELETED



Especifica si Visual FoxPro procesará o no los registros marcados para su eliminación, y si estarán disponibles o no para utilizarse en otros comandos.

SET DELETED ON | OFF

Parámetros

ON

Especifica que los comandos que realizan operaciones en registros (incluidos los registros de las tablas relacionadas) utilizando un alcance pasen por alto los registros marcados para su eliminación.

OFF

(Predeterminado) Especifica que puedan tener acceso a los registros marcados para eliminación los comandos que actúan sobre registros (incluidos los registros de las tablas relacionadas), utilizando un alcance.

4.4.4 SET EXCLUSIVE

Especifica si Visual FoxPro abre los archivos de tabla para uso exclusivo o compartido en una red.

SET EXCLUSIVE ON | OFF

Parámetros

ON

Al abrir una tabla para uso exclusivo se asegura que otros usuarios no puedan cambiar el archivo. Algunos comandos no pueden ejecutarse a menos que se abra una tabla para uso exclusivo. Estos comandos son INSERT, INSERT BLANK, MODIFY STRUCTURE, PACK y ZAP.

OFF

(El valor predeterminado para una sesión de datos privada.) Permite que cualquier usuario de la red pueda compartir y modificar una tabla abierta en red.

Para obtener información adicional acerca del bloqueo de registros y archivos, y el uso compartido de tablas en una red, vea Programar para acceso compartido.

Observaciones

Al cambiar la configuración de SET EXCLUSIVE no cambiará el estado de las tablas abiertas previamente. Por ejemplo, si una tabla se abre con SET EXCLUSIVE ON y SET EXCLUSIVE se cambia posteriormente a OFF, la tabla mantendrá su estado de uso exclusivo.

SET EXCLUSIVE tiene como alcance la sesión actual de datos.

ANEXOS

1. LISTADO DE OPERADORES

1.1. Numérico

Operador	Acción	Código
()	Subexpresiones de grupo	<pre>(4-3) * (12/nVar2)</pre>
** , ^	Potenciación	<pre>? 3 ** 2 ? 3 ^ 2</pre>
*, /	Multiplicación y división	<pre>? 2 * 7 ? 14 / 7</pre>
%	Módulo (resto)	<pre>? 15 % 4</pre>
+, -	Suma y resta	<pre>? 4 + 15</pre>

1.2. Cadena

Puede agrupar y comparar datos de caracteres con los operadores para caracteres +, -, y \$. En la siguiente tabla se muestran los operadores de expresión de caracteres en orden de prioridad.

Operadores para el tipo Character

Operador	Acción	Código
+	Concatenación. Combina dos cadenas, una cadena y un campo, o una cadena y una variable.	<pre>? 'Good ' + 'morning'</pre>
-	Concatenación. Quita los espacios finales del elemento anterior al operador y luego combina dos elementos.	<pre>? customer.first - customer.last</pre>
\$	Comparación. Busca una expresión de caracteres dentro de otra.	<pre>? 'father' \$ 'grandfather' ? 'Main' \$ customer.address</pre>

1.3. Relacionales

Operador	Acción	Código
<	Menor que	? 23 < 54
>	Mayor que	? 1 > 2
=	Igual que	? cVar1 = cVar
<>, #, !=	Distinto de	? .T. <> .F.
<=	Menor o igual que	? {^1998/02/16} <= {^1998/02/16}
>=	Mayor o igual que	? 32 >= nHisAge
==	Comparación de cadenas de caracteres	? status == "Open"

1.4 Lógicos

Operadores para el tipo Logical

Operador	Acción	Código
()	Grupos de expresiones	cVar AND (cVar2 AND cVar3)
NOT, !	Lógico negativo	IF NOT cVarA = cVarB IF ! nVar1 = nVar2
AND	AND lógico	lVar0 AND lVar9
OR	OR lógico inclusivo	lVarX OR lVarY

2.Código ASCII

El código ASCII

sigla en inglés de American Standard Code for Information Interchange
(Código Estadounidense Estándar para el Intercambio de Información)

Tomada del sitio:

tecnologiaeinformaticaa.es.tl

[illegible]

REFERENCIAS WEB

¿Qué es el Visual Fox?

https://es.wikipedia.org/wiki/Visual_FoxPro

Sobre DBMS

<http://www.mastermagazine.info/termino/4544.php>

Bases de datos y tablas

<http://www.tamps.cinvestav.mx/~jtj/courses/dbs/slides/chapter1dbs.pdf>

Importancia de la capacitación

<http://www.pymempresario.com/2012/07/la-importancia-de-la-capacitacion/>

Comandos VFP - MSDN

[https://msdn.microsoft.com/es-es/library/cc483355\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/cc483355(v=vs.71).aspx)

Ejemplo de Concatenar en VFP

<http://www.lawebdelprogramador.com/>

El código ASCII

<http://www.elcodigoascii.com.ar/codigos-ascii/letra-e-mayuscula-codigo-ascii-69.html>