



?FAC1:FechaEmission{PROP:Touched} = TRUE

Esta sentencia hace que cuando te posicionas sobre un campo que tiene un valor cargado y no lo modificas, el caso de dar enter y seguir, fuerce a hacer todas las validaciones.

! Valido Fecha Emission

```
IF FAC1:FechaEmission Not > 0
FAC1:FechaEmission = Today() ← Si el operador no ingresó una fecha le cargo la fecha de windows. (de la pc)
```

```
IF FAC1:FechaEmission > Today() ← Si la fecha ingresada es posterior a la fecha de la pc muestra un cartel y regresa el foco al campo FAC1:FechaEmission
MESSAGE('Fecha Posterior a La Actual!!!.....'&Format(Today(),@d06b),'Mensaje del Sistema',ICON:Exclamation)
Select(?FAC1:FechaEmission)
CYCLE
```

```
End
Display ← Refresco la pantalla
```

! Leo AAFact01 para ver si existe

```
IF FAC1:Numero > 0 AND SELF:REQUEST = INSERTRECORD
AAF:Tipcom = FAC1:Tipcom
AAF:Letra = FAC1:Letra
AAF:Puesto = FAC1:Puesto
AAF:Numero = FAC1:Numero
IF Not Access:AAFact01.Fetch(AAF:KeyClave) ← Leo el archivo AAFact01 y SI NO EXISTE no hace nada, de lo contrario muestra el cartel con la razonsocial que se confeccionó
MESSAGE('Ya Ingresó Ese Comprobante.....'&AAF:RazonSocial,'Mensaje del Sistema',ICON:Exclamation)
Select(?FAC1:CodCliente) ← Regresa el cursor al Codido de Cliente
CYCLE
END
END
```

```

? Acá es el caso que tengo un archivo padre y otro hijo. Es el caso de Cabecera de Factura y detalle de los Items
? Están relacionados por un único campo que se llama VectorUta. En el caso que uses mysql, sql| u otra BD es necesario que todos los archivos
? Tengan una clave única y primary key
? Yo tengo en la cabecera la clave FAC1:VectorUta que es autonumber y Primary Key
? El detalle tiene una clave FAC2:KeyVectorUta que es foranea y está relacionado de Muchos a Uno y debe existir en la Cabecera
?

FAC2:VectorUta    =    FAC1:VectorUta    ? Muevo el campo que es una clave
SET (FAC2:KeyVectorUta,FAC2:KeyVectorUta) ? Ordeno el archivo por esta clave y me posiciono sobre el registro. Se va a posicionar sobre el mas cercano,
                                           ? en caso de no existir se posicionará sobre el inmediato superior
LOOP UNTIL Access:AFact002.Next() or FAC2:VectorUta <> FAC1:VectorUta ? Leo secuencialmente mientras el campo FAC2:VectorUta sea igual a FAC1:VectorUta
? Acá va el código que necesites.
? Sumar importes, items, actualizar existencias, etc

END ? El loop debe terminar con un END

```

Igual que en el caso anterior

```

!
Clear(MTR:Record)           ? Limpio el registro del archivo
MTR:VectorUta    =    FAC1:VectorUta    ? Muevo el campo
Set(MTR:KeyVectorUta,MTR:KeyVectorUta) ? Ordeno y me posiciono
Loop Until Access:MTRibuto.Next() or MTR:VectorUta <> FAC1:VectorUta ? Leo mientras el campo sea igual
    Delete(MTRibuto)        ? Borro
    IF ErrorCode()         ? Controlo el error
        MESSAGE('No Borra MTRibuto.. ', 'Mensaje del Sistema',ICON:Exclamation)
    END
END

? Lectura de un archivo por una clave
AC1:CodCliente    =    FAC1:CodCliente    ? Mueve el campo
IF Access:Accte001.Fetch(AC1:KeyCodCliente) ? Lee por clave AC1:KeyCodCliente si existe continúa
    MESSAGE('Cliente Inexistente!!!!.....', 'Mensaje del Sistema',ICON:Exclamation) ? Si no existe muestra el cartel y manda el cursor a CodCliente
    Select(?FAC1:CodCliente) ? Acá no hace falta el "Cycle" por que no hay que interrumpir ningún LOOP
END

```

! Definir una Queue

! Una queue es una tabla que se carga en memoria y le podes agregar registros, modificar, borrar y leer

! La podes definir en Globales (no es recomendado) y en la local data

! Estas lineas siguientes las podes copiar tal cual en Local data y te vas a ver la definición de una queue

Procedure Afact001:Ficha\_Consulta Properties



Procedure Name: Afact001:Ficha\_Consulta  
 Template: Window(ABC)  
 Description: Ficha de Afact001  
 Category: Form  
 Prototype:  
 Module Name: cross363.clw  
☐ Declare Globally  
 Parameters:  
 Return Value:

Window Behavior  
 Listbox Styles  
 BIND Fields and Procedures

Record Validation  
 Control Value Validation Conditions..  
☒ Validate when the control is Accepted  
☒ Validate during Non-Stop Select  
☒ Color Fields rather than Selection  
 Field Color when Invalid COLOR:Red  
 Prompt Color when Invalid COLOR:Red  
☐ Show Message when fields are Invalid  
 Message to Display The values of Colored Field  
 Control to place Message in  
 Do Not Validate...

Save Button Properties  
 Allow: ☒ Inserts ☐ Changes ☐ Deletes  
 Field Priming on Insert Messages and Titles

☒ Tables  
☒ Window  
☐ Report  
☒ Data  
☒ Procedures  
☒ Embeds  
☐ Formulas  
☒ Extensions

OK

Cancel

Ingresar acá y al final  
colocar las líneas:

```
QHijo  QUEUE,PRE(QH)
NroInterno  LONG
CodArticulo STRING(20)
Denominacion STRING(60)
CtaContable DECIMAL(6,0)
Cantidad    DECIMAL(12,2)
PrecioUnitario DECIMAL(12,3)
ImporteTotal DECIMAL(12,2)
END
```

La podes cargar ingresando  
en DATA e indicarle que es  
una Queue y el Prefijo que  
elijas.

Vamos a cargar una queue con los items de una compra

```
I# = 0 ! Defino una variable intrínseca V# Almacena un decimal V" Almacena un string
      ! Este tipo de variables solo están activas y disponibles en esta rutina
COM2:VectorCpra = COM1:VectorCpra
SET (COM2:KeyVectorCpra, COM2:KeyVectorCpra)
LOOP UNTIL Access:Acompr02.Next() OR COM2:VectorCpra <> COM1:VectorCpra
    I# += 1 ! Sumo 1
    QH:NroInterno = COM2:NroInterno
    QH:CodArticulo = COM2:CodArticulo
    QH:Denominacion = COM2:Denominacion
    QH:Cantidad = COM2:Cantidad
    QH:PrecioUnitario = COM2:PrecioUnitario
    QH:ImporteTotal = COM2:ImporteTotal

    ADD (QHijo, I#) ! Agrego un registro
    IF ERRORCODE() THEN STOP (ERROR()). ! Controlo el error
END ! Loop
! *****
    ADD (QHijo, I#) ! Agrego un registro
    IF ERRORCODE() THEN STOP (ERROR()). ! Controlo el error

    PUT (QHijo, I#) ! Modifico un registro (Hago un update)
    IF ERRORCODE() THEN STOP (ERROR()). ! Controlo el error

    DELETE (QHijo, I#) ! Borro un registro
    IF ERRORCODE() THEN STOP (ERROR()). ! Controlo el error

    GET (QHijo, I#) ! Leo un registro en la posición I#
    IF ERRORCODE() THEN STOP (ERROR()). ! Controlo el error
! *****

! Imprimir desde una Queue
Loop I#= 1 to Records (QHijo) ! Hago un loop desde I# = 1 hasta la cantidad de registros que tiene la queue
    Get (QHijo, I#) ! Leo en esa posición
    Print (RPT:Detail) ! Imprimo un detail de un reporte
End ! Los campos de una Queue se pueden imprimir.
```