FExemplos - Progress Básico:

INSERT

======

/\* exemplo 1 \*/

insert Customer.

/\* exemplo 2 \*/

insert SalesRep with side-labels.

insert SalesRep with side-labels 2 columns.

Fazer -> RefCall (CustNum=22, SalesRep=bbb)

CREATE e UPDATE

===============

/\* exemplo 1 \*/

create Customer.

update Customer.

/\* exemplo 2 \*/

create SalesRep.

update SalesRep with side-labels 2 columns.

Fazer -> RefCall (CustNum=20,SalesRep=bbb)

DISPLAY

=======

/\* exemplo 1 \*/

find first Customer.

display Customer.

display Customer with side-labels 1 column width 100.

disp Customer except SalesRep.

Fazer -> RefCall

/\* exemplo 2 \*/

FIND FIRST Customer WHERE CustNum > 99999 NO-ERROR.

IF AVAILABLE Customer THEN DO:

DISPLAY Customer.CustNum

Customer.SalesRep FORMAT "x(1)"

Customer.Address

Customer.Address2

Customer.Balance

Customer.City

Customer.Comments

Customer.Contact

Customer.Country

Customer.CreditLimit

Customer.Discount

Customer.Name

Customer.Phone

Customer.PostalCode

Customer.Terms.

update Customer.Address

Customer.Address2

Customer.Balance

Customer.City

Customer.Comments

Customer.Contact

Customer.Country

Customer.CreditLimit

Customer.Discount

Customer.Name

Customer.Phone

Customer.PostalCode

Customer.Terms

with side-labels 2 COLUMNS.

END.

ELSE DO:

MESSAGE "registro nao encontrado".

END.

SET e PROMPT-FOR

================

/\* exemplo 1 \*/

create SalesRep.

set SalesRep.

/\* exemplo 2 \*/

prompt-for SalesRep.SalesRep with side-labels.

find SalesRep using SalesRep.SalesRep no-lock no-error.

if avail SalesRep then

disp SalesRep with 1 col side-labels.

Fazer -> State

/\* exemplo 3 \*/

DEF VAR i-CustNum AS integer NO-UNDO.

PROMPT-FOR i-CustNum.

FIND FIRST Customer

WHERE Customer.CustNum = INPUT i-CustNum

NO-LOCK NO-ERROR.

IF NOT AVAIL Customer THEN DO:

FIND FIRST Customer EXCLUSIVE-LOCK.

ASSIGN Customer.CustNum = INPUT i-CustNum.

END.

ELSE DO:

MESSAGE "ja existe registro com esse codigo '"

i-CustNum

"'. tente novamente"

VIEW-AS ALERT-BOX.

END.

/\* exemplo 4 \*/

DEF VAR codigo LIKE SalesRep.SalesRep NO-UNDO.

prompt-for codigo with side-labels.

ASSIGN codigo = INPUT codigo.

find SalesRep WHERE SalesRep.SalesRep = codigo no-lock no-error.

/\*

find SalesRep

WHERE SalesRep.SalesRep = SalesRep.SalesRep

no-lock no-error.

\*/

/\*

find SalesRep using SalesRep.SalesRep no-lock no-error.

\*/

if avail SalesRep then

disp SalesRep with 1 col side-labels.

ASSIGN e DELETE

===============

/\* exemplo 1 \*/

create SalesRep.

assign SalesRep.SalesRep = "zzz".

disp SalesRep.SalesRep.

delete SalesRep.

disp SalesRep.SalesRep.

Fazer -> RefCall (CustNum=25,SalesRep=bbb)

/\* exemplo 2 \*/

create SalesRep.

assign SalesRep.SalesRep = "zzz".

disp SalesRep.SalesRep.

delete SalesRep.

/\* PAUSE 5. \*/

FIND FIRST SalesRep

WHERE SalesRep.SalesRep = "bbb"

NO-LOCK NO-ERROR.

IF AVAIL SalesRep THEN DO:

disp SalesRep.SalesRep.

MESSAGE "registro encontrado" VIEW-AS ALERT-BOX.

END.

ELSE DO:

MESSAGE "o vendedor ZZZ nao existe mais"

VIEW-AS ALERT-BOX.

MESSAGE "teste de escrita" VIEW-AS ALERT-BOX.

END.

REPEAT

======

/\* exemplo 1 \*/

repeat:

insert Customer with 1 column.

end.

/\* exemplo 2 \*/

repeat:

create SalesRep.

update SalesRep.

end.

Fazer -> State

FOR EACH

========

/\* exemplo 1 \*/

for each Order no-lock:

disp Order.

for each OrderLine no-lock

where OrderLine.OrderNum = Order.OrderNum:

disp OrderLine.

end.

end.

/\* exemplo 2 \*/

for each SalesRep no-lock:

disp SalesRep.

end.

Fazer -> RefCall

/\* exemplo 3 \*/

for each Order no-lock

WHERE (Order.OrderNum > 30 AND Order.OrderNum < 110)

AND Order.CustNum = 40:

disp Order.OrderNum.

find Customer of Order no-lock no-error.

if avail Customer THEN DO:

disp Customer.CustNum

Customer.Name.

FIND State

WHERE State.State = Customer.State

/\*

AND State.Region = "aaa"

\*/

NO-LOCK NO-ERROR.

IF AVAIL State THEN DO:

DISP State.StateName.

END.

END.

for each OrderLine of Order no-lock:

disp OrderLine.

end.

end.

/\* exemplo 4 \*/

FOR EACH Customer:

DISP Customer.CustNum .

DISP Customer.Name

Customer.CreditLimit.

IF Customer.CreditLimit > 15000 THEN

NEXT.

IF Customer.CreditLimit = 10000 THEN

LEAVE.

UPDATE Customer.Name.

/\*

IF Customer.CreditLimit > 15000 THEN DO:

UPDATE Customer.Name.

MESSAGE "nome alterado".

END.

\*/

END.

/\* exemplo 5 \*/

for each SalesRep no-lock:

disp SalesRep.

end.

/\* exemplo 6 \*/

/\*

for each Order no-lock:

\*/

FOR EACH Order,

EACH Customer

WHERE Customer.CustNum = Order.CustNum NO-LOCK:

disp Order.

/\*

FIND Customer OF Order NO-LOCK NO-ERROR.

FIND Customer

WHERE Customer.CustNum = Order.CustNum NO-LOCK NO-ERROR.

\*/

DISP Customer.Name.

for each OrderLine no-lock

where OrderLine.OrderNum = Order.OrderNum:

disp OrderLine.

end.

end.

/\* exemplo 7 \*/

FOR EACH Customer:

DISP Customer.CustNum

Customer.Name.

/\*

ASSIGN Customer.Name = Customer.Name + " 1".

\*/

END.

/\* exemplo 8 \*/

DEF VAR l-resp AS LOGICAL

LABEL "Continua a incluir?" NO-UNDO.

REPEAT WITH 1 COLUMN:

create SalesRep.

update SalesRep.

UPDATE l-resp.

IF l-resp = NO THEN DO:

LEAVE.

END.

end.

DO

==

/\* exemplo 1 \*/

def var i-cont as integer no-undo.

do i-cont = 1 to 10 with frame f-pri:

disp i-cont with frame f-pri.

end.

/\* exemplo 2 \*/

do i-cont = 100 to 1 by -2:

disp i-cont with frame f-dad down.

down with frame f-dad.

end.

/\* exemplo 3 \*/

def var i-cont as integer no-undo.

do i-cont = 1 to 10 BY 2 WITH FRAME f-x:

disp i-cont .

end.

PAUSE.

do i-cont = 10 to 1 BY -1 WITH FRAME f-y:

disp i-cont .

/\*

PAUSE .5.

\*/

end.

FIND

====

/\* exemplo 1 \*/

for each Order no-lock:

disp Order.OrderNum.

find Customer of Order no-lock no-error.

if avail Customer then

disp Customer.CustNum

Customer.Name.

for each OrderLine of Order no-lock:

disp OrderLine.

end.

end.

Fazer -> State

Begins e Matches

================

/\* exemplo 1 \*/

FOR EACH Customer

/\*

WHERE Customer.Name BEGINS "bi"

\*/

WHERE Customer.Name MATCHES "\*ba\*"

USE-INDEX Name:

/\*

BY Balance DESCENDING

BY CustNum:

\*/

DISPLAY Customer.Balance

Customer.Name

Customer.CustNum

Customer.Phone.

END.

CONTAINS

========

/\* exemplo 1 \*/

FOR EACH Item

WHERE Item.ItemName CONTAINS "ball":

DISP Item.ItemNum

Item.ItemName

Item.Price.

END.

FOR EACH Item

WHERE Item.ItemName CONTAINS "bal":

DISP Item.ItemNum

Item.ItemName

Item.Price.

END.

BUFFER

======

/\* exemplo 1 \*/

def buffer bf-cust for Customer.

find first Customer no-lock no-error.

disp Customer.CustNum Customer.Name.

find first bf-cust

where bf-cust.CustNum > Customer.CustNum

no-lock no-error.

disp bf-cust.CustNum bf-cust.Name.

Fazer -> SalesRep

/\* exemplo 2 \*/

def buffer bf-cust for Customer.

def buffer bf-cust2 for Customer.

find first Customer no-lock no-error.

find first bf-cust

where bf-cust.CustNum > Customer.CustNum

no-lock no-error.

find first bf-cust2

where bf-cust2.CustNum > bf-cust.CustNum

EXCLUSIVE-LOCK no-error.

ASSIGN bf-cust2.Name = Customer.Name.

disp Customer.CustNum Customer.Name

bf-cust.CustNum bf-cust.Name

bf-cust2.CustNum bf-cust2.Name

WITH 1 COLUMN.

QUERY

=====

/\* exemplo 1 \*/

def query q-SalesRep for SalesRep.

open query q-SalesRep

for each SalesRep where SalesRep.SalesRep > "A".

get first q-SalesRep.

repeat:

if not avail SalesRep then

leave.

disp SalesRep.RepName.

get next q-SalesRep.

end.

/\* exemplo 2 \*/

DEF VAR l-resp AS LOGICAL FORMAT "Avanca/Retorna"

INITIAL YES NO-UNDO.

def query q-SalesRep for SalesRep.

open query q-SalesRep

for each SalesRep where SalesRep.SalesRep > "A".

get first q-SalesRep.

repeat:

if not avail SalesRep THEN DO:

leave.

END.

disp SalesRep.SalesRep SalesRep.RepName.

UPDATE l-resp.

IF l-resp = yes THEN

get next q-SalesRep.

ELSE

get PREV q-SalesRep.

end.

VARIAVEIS e MESSAGE

===================

/\* exemplo 1 \*/

def var i-x as integer no-undo.

def var c-nome as char no-undo label "Nome" init "Fulano".

do i-x = 5 to 1 by -1:

disp i-x.

update c-nome.

end.

message "O ultimo nome digitado foi " c-nome skip(2)

"Até a proxima"

view-as alert-box information.

Fazer -> Solicitar que façam alguns Messages com Error, Question, Warning

IF

==

/\* exemplo 1 \*/

def var l-cont as logical no-undo.

def var c-nome as char no-undo label "Nome".

repeat:

update c-nome.

if c-nome = "" then do:

message "O nome esta em branco, deseja sair ?" update l-cont

view-as alert-box question buttons yes-no-cancel.

if l-cont = yes then

leave.

end.

end.

Fazer -> Perguntar a senha, se a mesma for <> "Abacaxi" mostrar

mensagem de erro e perguntar a senha novamente.

CASE

====

/\* exemplo 1 \*/

def var c-senha as char no-undo label "senha".

repeat:

update c-senha.

case c-senha:

when "sai" then leave.

when "secreta" then do:

message "senha ok !" view-as alert-box.

leave.

end.

otherwise do:

message "senha errada, tente novamente !" view-as alert-box.

end.

end case.

end.

FRAME

=====

/\* exemplo 1 \*/

def frame f-cust

Customer.CustNum at 10

Customer.Name at 10

Customer.SalesRep at 10

with side-labels three-d title "Manutencao de Clientes"

view-as dialog-box.

for each Customer exclusive-lock:

update Customer.CustNum

Customer.Name

Customer.SalesRep with frame f-cust.

end.

Fazer -> frames down, side-label, no-labels, stream-io, etc.

Manipulação de Frames

=====================

/\* exemplo 1 \*/

def frame f-cust

Customer.CustNum at 10

Customer.Name at 10

Customer.SalesRep at 10

with 2 down side-labels title "Manutencao de Clientes"

view-as dialog-box.

view frame f-cust.

pause.

disp 5 @ Customer.CustNum with frame f-cust.

down with frame f-cust.

pause.

disp 10 @ Customer.CustNum with frame f-cust.

down with frame f-cust.

pause.

hide frame f-cust no-pause.

pause.

INCLUDE

=======

/\* exemplo 1 \*/

OBS: usar debug

{i-teste.i &tab=SalesRep &atr1=SalesRep &atr2=RepName}

for each {&tab} no-lock:

&if "{&atr1}" <> "" &then

disp {&tab}.{&atr1}.

&endif

&if "{&atr2}" <> "" &then

disp {&tab}.{&atr2}.

&endif

end.

OUTPUT

======

/\* exemplo 1 \*/

def stream s1.

def stream s2.

output stream s1 to c:\tmp\arq1.txt.

output stream s2 to c:\tmp\arq2.txt.

for each Order no-lock:

disp stream s1 Order with down stream-io.

end.

for each OrderLine no-lock:

disp stream s2 OrderLine with down stream-io.

end.

output stream s1 close.

output stream s2 close.

/\* exemplo 2 \*/

def stream s1.

def stream s2.

output stream s1 to c:/tmp/arq1.txt.

output stream s2 to c:/tmp/arq2.txt.

DEF FRAME f-Order

Order.OrderNum

Order.CustNum

Customer.Name

Order.SalesRep

SalesRep.RepName

Order.OrderDate

WITH DOWN STREAM-IO NO-ATTR-SPACE WIDTH 130.

for each Order no-lock:

disp stream s1

Order.OrderNum

Order.CustNum

Order.SalesRep

Order.OrderDate with FRAME f-Order.

FIND Customer

WHERE Customer.CustNum = Order.CustNum

NO-LOCK NO-ERROR.

IF AVAIL Customer THEN

DISP STREAM s1 Customer.Name WITH FRAME f-Order.

FIND SalesRep OF Order NO-LOCK NO-ERROR.

IF AVAIL SalesRep THEN

DISP STREAM s1 SalesRep.RepName WITH FRAME f-Order.

end.

for each OrderLine no-lock:

disp stream s2 OrderLine with down stream-io.

end.

output stream s1 close.

output stream s2 close.

OS-COMMAND NO-WAIT VALUE("notepad c:\tmp\arq1.txt").

OS-COMMAND NO-WAIT VALUE("notepad c:\tmp\arq2.txt").

BREAK BY

========

/\* exemplo 1 \*/

for each Order no-lock

break by Order.CustNum

by Order.OrderNum:

if first-of(Order.CustNum) then do:

find Customer of Order no-lock no-error.

if avail Customer then

disp Customer.CustNum Customer.Name with frame f-x.

end.

disp Order except Order.CustNum iwth frame f-x.

end.

/\* exemplo 2 \*/

ASSIGN CURRENT-WINDOW:WIDTH=200.

DEF VAR icont AS INTEGER NO-UNDO.

DEF VAR msg AS CHAR NO-UNDO FORMAT "x(10)".

DEF STREAM sRelat.

OUTPUT STREAM sRelat TO c:\tmp\relat.txt.

DEF FRAME f-cab HEADER

"Relatorio de pedidos"

WITH CENTERED.

VIEW STREAM sRelat FRAME f-cab.

for each Order NO-LOCK

break by Order.CustNum

by Order.OrderNum:

if first-of(Order.CustNum) then do:

find Customer of Order no-lock no-error.

if avail Customer THEN DO:

disp STREAM sRelat

Customer.CustNum Customer.Name

with frame f-x.

END.

end.

disp STREAM sRelat Order

except Order.CustNum Order.instructions Order.Terms

with frame f-x DOWN WIDTH 150 STREAM-IO.

ASSIGN icont = icont + 1.

IF LAST-OF(Order.CustNum) THEN DO:

DISP STREAM sRelat "SubTotal=" + String(icont) @ msg

WITH FRAME f-x.

DOWN 1 STREAM sRelat WITH FRAME f-x.

ASSIGN icont = 0.

END.

end.

OUTPUT STREAM sRelat CLOSE.

OS-COMMAND NO-WAIT VALUE("notepad c:\tmp\relat.txt").

/\* exemplo 3 \*/

ASSIGN CURRENT-WINDOW:WIDTH=200.

DEF VAR icont AS INTEGER NO-UNDO.

DEF VAR msg AS CHAR NO-UNDO FORMAT "x(10)".

DEF STREAM sRelat.

OUTPUT STREAM sRelat TO c:\tmp\relat.txt PAGE-SIZE 30.

DEF FRAME f-cab HEADER

"Relatorio de pedidos"

"Pag.:" TO 98

PAGE-NumBER(sRelat) FORMAT "99999" TO 110

WITH CENTERED PAGE-TOP WIDTH 150.

DEF FRAME f-rodape HEADER

FILL("=", 150) FORMAT "x(130)"

WITH PAGE-BOTTOM WIDTH 152 STREAM-IO.

VIEW STREAM sRelat FRAME f-cab.

VIEW STREAM sRelat FRAME f-rodape.

for each Order NO-LOCK

break by Order.CustNum

by Order.OrderNum:

if first-of(Order.CustNum) then do:

find Customer of Order no-lock no-error.

if avail Customer THEN DO:

disp STREAM sRelat

Customer.CustNum Customer.Name

with frame f-x.

END.

end.

disp STREAM sRelat Order

except Order.CustNum Order.instructions Order.Terms

with frame f-x 10 DOWN WIDTH 150 STREAM-IO.

ASSIGN icont = icont + 1.

IF LAST-OF(Order.CustNum) THEN DO:

DISP STREAM sRelat "SubTotal=" + String(icont) @ msg

WITH FRAME f-x.

DOWN 1 STREAM sRelat WITH FRAME f-x.

ASSIGN icont = 0.

END.

end.

OUTPUT STREAM sRelat CLOSE.

OS-COMMAND NO-WAIT VALUE("notepad c:\tmp\relat.txt").

BOTOES

======

/\* exemplo 1 \*/

def button bt-ok label "&OK" size 10 by 1 auto-go.

def button bt-detalhe label "&Detalhe" size-10 by 1.

def button bt-cancela label "&Cancela" size-10 by 1 auto-endkey.

update bt-ok bt-detalhe bt-cancela.

Fazer -> Botão "Processa"

/\* exemplo 2 \*/

def button bt-ok label "&OK" size 10 by 1 auto-go.

def button bt-detalhe label "&Detalhe" SIZE 10 by 1.

def button bt-cancela label "&Cancela" SIZE 10 by 1 auto-endkey.

DEF FRAME f-dad

bt-ok AT ROW 1 COL 10

bt-detalhe AT ROW 3 COL 20

bt-cancela AT ROW 5 COL 40

WITH SIZE 50 BY 10.

ON CHOOSE OF bt-detalhe DO:

MESSAGE "botao pressionado" VIEW-AS ALERT-BOX.

END.

update bt-ok bt-detalhe bt-cancela WITH FRAME f-dad.

FILL-IN

=======

/\* exemplo 1 \*/

def var c-nome as char label "Nome" no-undo.

update c-nome with side-labels.

Fazer -> Fill-in "Contato"

/\* exemplo 2 \*/

def var c-nome as char label "Nome" no-undo

VIEW-AS EDITOR SIZE 50 BY 3 SCROLLBAR-VERTICAL.

update c-nome VIEW-AS FILL-IN LABEL "dados"

with side-labels.

TOGGLE-BOX

==========

/\* exemplo 1 \*/

def var l-autom as logical label "Possui Automovel" view-as toggle-box.

def var l-casa as logical label "Possui Casa Propria" view-as toggle-box.

update l-autom l-casa.

Fazer -> Toggle-box "Processa proximo Mes ?"

/\* exemplo 2 \*/

def var l-autom as logical label "Possui Automovel"

view-as toggle-box.

def var l-casa as logical label "Possui Casa Propria"

view-as toggle-box.

DEF BUTTON bt-canc LABEL "cancela" AUTO-ENDKEY.

DEF FRAME f-x

l-casa

l-autom SKIP(8)

bt-canc

WITH SIDE-LABELS SIZE 60 BY 10.

ON ENTRY OF l-autom DO:

DEF VAR lin AS INTEGER NO-UNDO.

ASSIGN lin = l-autom:ROW + 1.

IF lin > FRAME f-x:HEIGHT THEN

ASSIGN lin = 1.

ASSIGN l-autom:ROW = lin.

END.

ON value-changed OF l-autom DO:

MESSAGE l-autom:LABEL l-autom:SCREEN-VALUE l-autom:CHECKED

VIEW-AS ALERT-BOX.

END.

ENABLE ALL WITH FRAME f-x.

WAIT-FOR CHOOSE OF bt-canc.

/\*

WAIT-FOR WINDOW-CLOSE OF CURRENT-WINDOW.

\*/

MESSAGE "depois" SKIP(1)

l-autom:LABEL l-autom:SCREEN-VALUE l-autom:CHECKED

VIEW-AS ALERT-BOX.

/\*

update l-autom l-casa WITH FRAME f-x.

\*/

RADIO-SET

=========

/\* exemplo 1 \*/

def var c-estcivil as char format "x(15)" label "Estado Civil" initial "Casado" no-undo

view-as radio-set radio-buttons "Solteiro", "Solteiro", "Casado", "Casado".

update c-estcivil.

Fazer -> Radio-Set "Titulo" com opcoes "Normal" e "Antecipação"

/\* exemplo 2 \*/

def var c-estcivil as char format "x(15)"

label "Estado Civil" initial "Casado" no-undo

view-as radio-set

radio-buttons "Solteiro", "Solteiro", "Casado", "Casado".

DEF FRAME f-rad

c-estcivil

WITH SIDE-LABELS.

ON VALUE-CHANGED OF c-estcivil DO:

IF c-estcivil:SCREEN-VALUE = "casado" THEN

MESSAGE "vc é casado" VIEW-AS ALERT-BOX.

MESSAGE "estado civil" c-estcivil:SCREEN-VALUE

VIEW-AS ALERT-BOX.

END.

update c-estcivil WITH FRAME f-rad.

SELECTION-LIST

==============

/\* exemplo 1 \*/

def var c-escolaridade as char label "Escolaridade" no-undo

view-as selection-list inner-chars 20 inner-lines 5

list-Items "1o. grau", "2o. grau", "3o. grau", "Pos-graduado".

Fazer -> SL-Classificacao = "Codigo", "Nome"

EDITOR

======

/\* exemplo 1 \*/

def var c-ender as char label "Endereco" no-undo

view-as editor size 40 by 3 scrollbar-vertical.

update c-ender.

Fazer -> c-carta size 70 by 10

COMBO-BOX

=========

/\* exemplo 1 \*/

def var c-uf as char format "x(10)" label "Estado" no-undo

view-as combo-box inner-lines 5 list-Items "SC","DF","RS","AM".

update c-uf.

Fazer -> Tipo de Pessoa = "Fisica", "Juridica"

IMAGEM

======

/\* exemplo 1 \*/

def image im-fundo image-file "c:\windows\nuvens.bmp" size 79 by 16.5.

def button bt-ok label "&OK" size 10 by 1 auto-go.

def frame f-x

im-fundo at row 1 col 1

bt-ok at row 3 col 2

with no-labels view-as dialog-box.

update bt-ok with frame f-x.

Fazer -> Imagem livre no HD

RECTANGLE

=========

/\* exemplo 1 \*/

def rectangle rt-mold size 79 by 16.5 edge-pixels 2.

def button bt-ok label "&OK" size 10 by 1 auto-go.

def frame f-x

rt-mold at row 1 col 1

bt-ok at row 3 col 2

with no-labels view-as dialog-box.

update bt-ok with frame f-x.

Fazer -> Retangulo qualquer

BROWSE

======

/\* exemplo 1 \*/

def query q-State for State.

def browse br-State query q-State disp

State.State

State.StateName

with separators size 40 by 4.

def frame f-x

br-State at 3

bt-ok at 3 space(5)

with side-labels view-as dialog-box three-d size 80 by 17.

open query q-State for each State.

update br-State bt-ok with frame f-x.

Solucao Exercicio

=================

def button bt-ok label "&OK" size 10 by 1 auto-go.

def button bt-detalhe label "&Detalhe" size-10 by 1.

def button bt-cancela label "&Cancela" size-10 by 1 auto-endkey.

def var c-nome as char label "Nome" no-undo.

def var l-autom as logical label "Possui Automovel" view-as toggle-box.

def var l-casa as logical label "Possui Casa Propria" view-as toggle-box.

def var c-estcivil as char format "x(15)" label "Estado Civil" initial "Casado" no-undo

view-as radio-set radio-buttons "Solteiro", "Solteiro", "Casado", "Casado".

def var c-escolaridade as char label "Escolaridade" no-undo

view-as selection-list inner-chars 20 inner-lines 5

list-Items "1o. grau", "2o. grau", "3o. grau", "Pos-graduado".

def var c-ender as char label "Endereco" no-undo

view-as editor size 40 by 3 scrollbar-vertical.

def var c-uf as char format "x(10)" label "Estado" no-undo

view-as combo-box inner-lines 5 list-Items "SC","DF","RS","AM".

def image im-fundo image-file "c:\windows\nuvens.bmp" size 79 by 16.5.

def rectangle rt-mold size 79 by 16.5 edge-pixels 2.

def query q-State for State.

def browse br-State query q-State disp

State.State

State.StateName

with separators size 40 by 4.

def frame f-dados

im-fundo at row 1 col 1

rt-mold at row 1 col 1

c-nome at row 2 col 3

c-estcivil at 3

c-escolaridade at 3

l-autom at 3

l-casa

c-ender at 3

c-uf at 3

br-State at 3

bt-ok at 3 space(5)

bt-detalhe space(5)

bt-cancela

with side-labels view-as dialog-box three-d size 80 by 17.

on choose of bt-detalhe do:

message c-nome view-as alert-box.

end.

open query q-State for each State.

enable all with frame f-dados.

wait-for go of frame f-dados.

TEMP-TABLE

==========

/\* exemplo 1 \*/

def temp-table tt-dados no-undo

field codigo as integer label "Codigo"

field nome as char format "x(20)" label "Nome"

index cod is primary unique codigo.

repeat:

insert tt-dados.

end.

for each tt-dados:

disp tt-dados.

end.

Fazer -> tt-State like State, incluir 2 estados e mostrar os

mesmos no final.

/\* exemplo 2 \*/

def temp-table tt-dados no-undo

field codigo as integer label "Codigo"

field nome as char format "x(20)" label "Nome"

index cod is primary unique codigo.

repeat:

insert tt-dados.

end.

for each tt-dados:

disp tt-dados WITH SIDE-LABELS.

end.

/\* exemplo 3 \*/

DEF TEMP-TABLE tt-cust NO-UNDO

FIELD CustNum LIKE Customer.CustNum

FIELD cName LIKE Customer.Name

FIELD RepName LIKE SalesRep.RepName

FIELD repcod LIKE Customer.SalesRep

FIELD State LIKE Customer.State

FIELD StateName LIKE State.StateName

FIELD totPed AS DECIMAL LABEL "total dos pedidos"

INDEX myindice IS PRIMARY UNIQUE CustNum.

DEF VAR tot AS DECIMAL NO-UNDO.

FOR EACH Customer NO-LOCK:

FIND SalesRep OF Customer NO-LOCK NO-ERROR.

FIND State OF Customer NO-LOCK NO-ERROR.

CREATE tt-cust.

ASSIGN tt-cust.CustNum = Customer.CustNum

tt-cust.cName = Customer.Name

tt-cust.repcod = Customer.SalesRep

tt-cust.State = Customer.State.

IF AVAIL SalesRep THEN

ASSIGN tt-cust.RepName = SalesRep.RepName.

IF AVAIL State THEN

ASSIGN tt-cust.StateName = State.StateName.

ASSIGN tot = 0.

FOR EACH Order OF Customer NO-LOCK:

FOR EACH OrderLine OF Order NO-LOCK:

ASSIGN tot = tot

+ (OrderLine.Price \* OrderLine.Qty)

- OrderLine.Discount.

END.

END.

ASSIGN tt-cust.totPed = tot.

END.

ASSIGN CURRENT-WINDOW:WIDTH = 150.

FOR EACH tt-cust NO-LOCK:

DISP tt-cust WITH FRAME f-x DOWN WIDTH 149.

END.

PI, RUN e COMPILACAO

====================

/\* exemplo 1 \*/

for each Order no-lock:

disp Order.OrderNum.

run pi-processa.

end.

procedure pi-processa:

for each OrderLine of Order no-lock:

disp OrderLine.

end.

end procedure.

compile teste.p save into c:\tmp.

/\* exemplo 2 \*/

COMPILE c:\treinamento\teste.p

PREPROCESS c:\treinamento\teste\_pre.txt

XREF c:\treinamento\teste\_xref.txt

LISTING c:\treinamento\teste\_list.txt

SAVE.

/\* exemplo 3 \*/

/\* persistente.p \*/

/\*

RUN pi-calcula.

\*/

DEF VAR zzz AS CHAR NO-UNDO.

MESSAGE "estou aqui - " zzz VIEW-AS ALERT-BOX.

PROCEDURE pi-calcula:

ASSIGN zzz = "calcula".

MESSAGE "executando calcula - " zzz VIEW-AS ALERT-BOX.

END PROCEDURE.

PROCEDURE pi-resposta:

ASSIGN zzz = "resposta".

MESSAGE "executando resposta - " zzz VIEW-AS ALERT-BOX.

END PROCEDURE.

/\* Programa chamador \*/

DEFINE VARIABLE wProg AS HANDLE NO-UNDO.

RUN c:/treinamento/persistente.p PERSISTENT SET wProg.

RUN pi-calcula IN wProg.

RUN pi-resposta IN wProg.

RUN pi-calcula IN wProg.

DISP int(wProg).

/\*

DELETE PROCEDURE wProg.

\*/

DELETE OBJECT wProg.

/\* exemplo 4 \*/

DEFINE VARIABLE wProg AS HANDLE NO-UNDO.

DEFINE VARIABLE calc AS INTEGER NO-UNDO.

RUN c:/treinamento/persistente.p PERSISTENT SET wProg.

RUN pi-calcula IN wProg (INPUT 2, INPUT 3, OUTPUT calc).

DISP calc.

RUN pi-resposta IN wProg (INPUT-OUTPUT calc).

DISP calc.

RUN pi-calcula IN wProg (5, 5, OUTPUT calc).

DISP calc.

/\*

DISP int(wProg).

DELETE PROCEDURE wProg.

\*/

DELETE OBJECT wProg.

/\* exemplo 5 \*/

def var l-existe as logical no-undo.

for each Order no-lock:

disp Order.OrderNum.

run pi-processa (input Order.OrderNum, output l-existe).

end.

procedure pi-processa:

def input param p-OrderNum as integer no-undo.

def output param p-achou as logical no-undo init no.

for each OrderLine

where OrderLine.OrderNum = p-OrderNum no-lock:

disp OrderLine.

assign p-achou = yes.

end.

end procedure.

EXEMPLOS DIVERSOS DE WIDGETS

============================

/\* exemplo 1 \*/

def var c-escol as char label "Escolaridade" no-undo

view-as selection-list

inner-chars 20 inner-lines 8

list-Items "1o. grau", "2o. grau",

"3o. grau", "Pos-graduado".

DEF BUTTON bt-canc LABEL "cancela" SIZE 20 BY 2 AUTO-ENDKEY.

DEF VAR c-dados AS CHAR FORMAT "x(20)" NO-UNDO.

DEF FRAME f-x

c-escol

c-dados NO-LABEL SKIP(1)

bt-canc

WITH SIDE-LABELS SIZE 80 BY 10

VIEW-AS DIALOG-BOX TITLE "Escolaridade".

ON CHOOSE OF bt-canc DO:

MESSAGE "o botao cancela foi pressionado"

VIEW-AS ALERT-BOX.

END.

ON VALUE-CHANGED OF c-escol DO:

ASSIGN c-dados:SCREEN-VALUE = c-escol:SCREEN-VALUE.

END.

ENABLE c-escol bt-canc WITH FRAME f-x.

WAIT-FOR CHOOSE OF bt-canc IN FRAME f-x.

ASSIGN c-dados.

MESSAGE "valor de c-dados = " c-dados VIEW-AS ALERT-BOX.

/\* exemplo 2 \*/

def var c-uf as char format "x(10)" label "Estado" no-undo

view-as combo-box

inner-lines 5

list-Items "SC","DF","RS","AM".

DEF FRAME f-x

c-uf

WITH SIDE-LABELS.

VIEW FRAME F-X.

ASSIGN c-uf:LIST-ItemS = c-uf:LIST-ItemS + ",CE".

c-uf:ADD-LAST("PR") IN FRAME f-x.

update c-uf WITH FRAME f-x.

MESSAGE "vc selecionou " c-uf VIEW-AS ALERT-BOX.

/\* exemplo 3 \*/

def image im-fundo FILE "c:\windows\Cafezinho.bmp"

size 79 by 16.5.

def button bt-ok label "&OK"

IMAGE-UP FILE "c:\windows\Pescaria.bmp"

size 20 by 4 AUTO-GO.

def frame f-x

bt-ok at row 3 col 2

im-fundo at row 1 col 1

with no-labels view-as dialog-box

SIZE 80 BY 20.

update bt-ok with frame f-x.

/\* exemplo 4 \*/

def rectangle rt-mold size 59 by 5 edge-pixels 2 BGCOLOR 1.

def rectangle rt-mold2 size 59 by 5 edge-pixels 2.

def button bt-ok label "&OK" size 10 by 1 auto-go.

def frame f-x

rt-mold at row 1 col 1 SKIP(1)

rt-mold2

bt-ok at row 3 col 2

with no-labels view-as DIALOG-BOX SIZE 100 BY 20.

update bt-ok with frame f-x.

/\* exemplo 5 \*/

DEF BUTTON bt-ok LABEL "&ok" AUTO-GO.

def query q-State for State, Customer SCROLLING.

def browse br-State query q-State disp

State.State

State.StateName

Customer.Name

with separators size 60 by 10.

def frame f-x

br-State at 3

bt-ok at 3

with side-labels view-as dialog-box

three-d size 80 by 17.

ON VALUE-CHANGED OF br-State DO:

MESSAGE State.StateName Customer.CustNum

VIEW-AS ALERT-BOX.

END.

open query q-State for each State,

EACH Customer OF State.

update br-State bt-ok with frame f-x.

/\* exemplo 6 \*/

def button bt-ok label "&OK" size 10 by 1 auto-go.

def button bt-detalhe label "&Detalhe" SIZE 10 by 1.

def button bt-cancela label "&Cancela" SIZE 10 by 1 auto-endkey.

def var c-nome as char label "Nome" no-undo.

def var l-autom as logical label "Possui Automovel" view-as toggle-box.

def var l-casa as logical label "Possui Casa Propria" view-as toggle-box.

def var c-estcivil as char format "x(15)" label "Estado Civil" initial "Casado" no-undo

view-as radio-set radio-buttons "Solteiro", "Solteiro", "Casado", "Casado".

def var c-escolaridade as char label "Escolaridade" no-undo

view-as selection-list inner-chars 20 inner-lines 5

list-Items "1o. grau", "2o. grau", "3o. grau", "Pos-graduado".

def var c-ender as char label "Endereco" no-undo

view-as editor size 40 by 3 scrollbar-vertical.

def var c-uf as char format "x(10)" label "Estado" no-undo

view-as combo-box inner-lines 5 list-Items "SC","DF","RS","AM".

def image im-fundo file "c:\windows\Pescaria.bmp" size 79 by 16.5.

def rectangle rt-mold size 79 by 16.5 edge-pixels 2.

def query q-State for State.

def browse br-State query q-State disp

State.State

State.StateName

with separators size 40 by 4.

def frame f-dados

im-fundo at row 1 col 1

rt-mold at row 1 col 1

c-nome at row 2 col 3

c-estcivil at 3

c-escolaridade at 3

l-autom at 3

l-casa

c-ender at 3

c-uf at 3

br-State at 3

bt-ok at 3 space(5)

bt-detalhe space(5)

bt-cancela

with side-labels view-as dialog-box three-d size 90 by 20.

on choose of bt-detalhe do:

message c-nome view-as alert-box.

end.

open query q-State for each State.

enable all with frame f-dados.

wait-for go of frame f-dados.

/\* exemplo 7 \*/

DEFINE BUTTON bt-ok LABEL "ok" SIZE 20 BY 3.

DEFINE BUTTON bt-canc LABEL "canc" SIZE 20 BY 1.

DEF FRAME f-x

bt-ok AT 1

bt-canc AT 1

WITH NO-LABELS.

ON CHOOSE, mouse-select-dblclick OF bt-canc, bt-ok DO:

ASSIGN bt-ok:VISIBLE = FALSE.

END.

ENABLE ALL WITH FRAME f-x.

APPLY "choose" TO bt-canc IN FRAME f-x.

WAIT-FOR WINDOW-CLOSE OF CURRENT-WINDOW.

NEW SHARED e SHARED

===================

/\* exemplo 1 \*/

def new shared temp-table tt-dados no-undo

field codigo as integer

field nome as char.

create tt-dados.

assign tt-dados.codigo = 1

tt-dados.nome = "fulano".

run prog-b.

/\* prog-b \*/

def shared temp-table tt-dados no-undo

field codigo as integer

field nome as char.

for each tt-dados no-lock:

disp tt-dados.

end.

/\* exemplo 2 \*/

def new shared temp-table tt-dados no-undo

field codigo as integer

field nome as char.

create tt-dados.

assign tt-dados.codigo = 1

tt-dados.nome = "fulano".

run prog-b.p.

/\* prog-b \*/

def shared temp-table tt-dados no-undo

field codigo as integer

field nome as char.

create tt-dados.

assign tt-dados.codigo = 2

tt-dados.nome = "beltrano".

for each tt-dados no-lock:

disp tt-dados.

end.

INPUT, OUTPUT e INPUT-OUTPUT

============================

/\* exemplo 1 \*/

def var l-existe as logical no-undo.

for each Order no-lock:

disp Order.OrderNum.

run pi-processa (input Order.OrderNum, output l-existe).

end.

procedure pi-processa:

def input param p-OrderNum as integer no-undo.

def output param p-achou as logical no-undo init no.

for each OrderLine

where OrderLine.OrderNum = p-OrderNum no-lock:

disp OrderLine.

assign p-achou = yes.

end.

end procedure.

/\* exemplo 2 \*/

/\* persistente.p \*/

/\*

RUN pi-calcula.

\*/

DEF VAR zzz AS CHAR NO-UNDO.

MESSAGE "estou aqui - " zzz VIEW-AS ALERT-BOX.

PROCEDURE pi-calcula:

DEF INPUT PARAM p1 AS INTEGER NO-UNDO.

DEF INPUT PARAM p2 AS INTEGER NO-UNDO.

DEF OUTPUT PARAM p3 AS INTEGER NO-UNDO.

ASSIGN p3 = p1 \* p2

zzz = "calcula".

MESSAGE "executando calcula - " zzz p3 VIEW-AS ALERT-BOX.

END PROCEDURE.

PROCEDURE pi-resposta:

DEF INPUT-OUTPUT PARAM pcalc AS INTEGER NO-UNDO.

ASSIGN pcalc = pcalc \* 1000

zzz = "resposta".

MESSAGE "executando resposta - " pcalc VIEW-AS ALERT-BOX.

END PROCEDURE.

PARAM BUFFER

============

/\* exemplo 1 \*/

def buffer b-cust for Customer.

for each Customer no-lock:

run pi-processa(buffer Customer).

end.

procedure pi-processa:

def param buffer b-cust for Customer.

disp b-cust.CustNum

b-cust.Name.

end procedure.

PARAM TABLE

===========

/\* exemplo 1 \*/

def temp-table tt-dados no-undo like Customer.

run pi-processa(output table tt-dados).

for each tt-dados:

disp tt-dados.CustNum

tt-dados.Name.

end.

procedure pi-processa:

def output param table for tt-dados.

for each Customer no-lock:

create tt-dados.

buffer-copy Customer to tt-dados.

end.

end procedure.

EXPORT e IMPORT

===============

/\* exemplo 1 \*/

def temp-table tt-dados no-undo like Customer.

output to c:\tmp\exp.txt.

for each Customer no-lock:

export Customer.

end.

output close.

/\*

OS-COMMAND NO-WAIT value("notepad c:\tmp\exp.txt").

\*/

input from c:\tmp\exp.txt.

repeat:

create tt-dados.

import tt-dados.

end.

input close.

for each tt-dados:

disp tt-dados.CustNum

tt-dados.Name.

end.

/\* exemplo 2 - Export delimiter \*/

def temp-table tt-State like State.

output to c:\temp\teste.csv.

for each State:

export delimiter ";" State.

end.

output close.

input from c:\temp\teste.csv.

/\*cria os dados de acordo com o arquivo\*/

repeat:

create tt-State.

import delimiter ";" tt-State.

end.

input close.

/\*mostra dados\*/

for each tt-State no-lock:

disp tt-State.

end.

PARAM BUFFER

============

/\* exemplo 1 \*/

def buffer b-cust for Customer.

for each Customer no-lock:

run pi-processa(buffer Customer).

end.

procedure pi-processa:

def param buffer b-cust for Customer.

disp b-cust.CustNum

b-cust.Name.

end procedure.

PARAM TABLE

===========

/\* exemplo 1 \*/

def temp-table tt-dados no-undo like Customer.

run pi-processa(output table tt-dados).

for each tt-dados:

disp tt-dados.CustNum

tt-dados.Name.

end.

procedure pi-processa:

def output param table for tt-dados.

for each Customer no-lock:

create tt-dados.

buffer-copy Customer to tt-dados.

end.

end procedure.

ERROS

=====

/\* exemplo 1 \*/

do on error undo, leave

on stop undo, leave

on quit undo, leave:

run xxx.p.

end.

message "passei por aqui" view-as alert-box.

TRANSACAO

=========

/\* exemplo 1 \*/

def temp-table tt-dados like State.

do transaction:

repeat:

insert tt-dados.

end.

undo, leave.

end.

for each tt-dados:

disp tt-dados.

end.

/\* exemplo 2 \*/

bloco\_pai:

do on error UNDO bloco\_pai, LEAVE bloco\_pai

on stop undo, leave

on quit undo, leave:

/\*

QUIT.

\*/

bloco\_filho:

FOR EACH Customer EXCLUSIVE-LOCK

ON ERROR UNDO bloco\_filho, NEXT bloco\_filho:

DISP Customer.CustNum.

DELETE Customer NO-ERROR.

IF ERROR-STATUS:ERROR THEN DO:

MESSAGE "deu erro no codigo " Customer.CustNum

VIEW-AS ALERT-BOX.

END.

END.

message "Customers eliminados" view-as alert-box.

run x.p NO-ERROR.

IF ERROR-STATUS:ERROR THEN DO:

MESSAGE "nao conseguiu executar o programa x.p "

VIEW-AS ALERT-BOX.

END.

end.

message "passei por aqui" view-as alert-box.

/\*x.p - versao 1\*/

MESSAGE "1-" TRANSACTION VIEW-AS ALERT-BOX.

RUN pi-processa.

MESSAGE "2-" TRANSACTION VIEW-AS ALERT-BOX.

PROCEDURE pi-processa:

MESSAGE "3-dentro da pi - " TRANSACTION VIEW-AS ALERT-BOX.

FIND FIRST Customer EXCLUSIVE-LOCK NO-ERROR.

ASSIGN Customer.SalesRep = "BBB".

MESSAGE "4-dentro da pi - " TRANSACTION VIEW-AS ALERT-BOX.

END PROCEDURE.

/\*x.p - versao 2 \*/

MESSAGE "1-" TRANSACTION VIEW-AS ALERT-BOX.

RUN pi-processa.

IF RETURN-VALUE = "NOK" THEN DO:

MESSAGE "houve erro durante a execucao da PI" VIEW-AS ALERT-BOX.

END.

MESSAGE "2-" TRANSACTION VIEW-AS ALERT-BOX.

PROCEDURE pi-processa:

MESSAGE "3-dentro da pi - " TRANSACTION VIEW-AS ALERT-BOX.

DO TRANSACTION ON ERROR UNDO, RETURN "NOK":

FIND Customer

WHERE Customer.CustNum = 999453

EXCLUSIVE-LOCK NO-ERROR.

ASSIGN Customer.SalesRep = "BBB" NO-ERROR.

IF ERROR-STATUS:ERROR THEN DO:

MESSAGE "deu erro - " ERROR-STATUS:GET-MESSAGE(1)

VIEW-AS ALERT-BOX.

UNDO, RETURN "NOK".

END.

END.

/\*

IF Customer.SalesRep = "bbb" THEN

RETURN "NOK".

\*/

MESSAGE "4-dentro da pi - " TRANSACTION VIEW-AS ALERT-BOX.

RETURN "OK".

END PROCEDURE.

/\*x.p - versao 3 \*/

MESSAGE "1-" TRANSACTION VIEW-AS ALERT-BOX.

DO TRANSACTION:

FIND FIRST Order EXCLUSIVE-LOCK.

END.

RUN pi-processa.

IF RETURN-VALUE = "NOK" THEN DO:

MESSAGE "houve erro durante a execucao da PI" VIEW-AS ALERT-BOX.

END.

MESSAGE "2-" TRANSACTION VIEW-AS ALERT-BOX.

FOR EACH OrderLine OF Order TRANSACTION:

DISP OrderLine.

END.

PROCEDURE pi-processa:

MESSAGE "3-dentro da pi - " TRANSACTION VIEW-AS ALERT-BOX.

DO TRANSACTION ON ERROR UNDO, RETURN "NOK":

FIND Customer

WHERE Customer.CustNum = 999453

EXCLUSIVE-LOCK NO-ERROR.

ASSIGN Customer.SalesRep = "BBB" NO-ERROR.

IF ERROR-STATUS:ERROR THEN DO:

MESSAGE "deu erro - " ERROR-STATUS:GET-MESSAGE(1)

VIEW-AS ALERT-BOX.

UNDO, RETURN "NOK".

END.

END.

/\*

IF Customer.SalesRep = "bbb" THEN

RETURN "NOK".

\*/

MESSAGE "4-dentro da pi - " TRANSACTION VIEW-AS ALERT-BOX.

RETURN "OK".

END PROCEDURE.

/\* exemplo 3 \*/

DEF BUFFER bf-cust FOR Customer.

FOR EACH Customer NO-LOCK:

IF Customer.CreditLimit = 15000 THEN DO TRANSACTION:

FIND bf-cust

WHERE ROWID(bf-cust) = ROWID(Customer)

EXCLUSIVE-LOCK NO-ERROR.

IF AVAIL bf-cust THEN DO:

ASSIGN bf-cust.SalesRep = "BBB".

END.

END.

END.

Bloqueios

=========

/\* exemplo 1 \*/

for each Customer no-lock:

update Customer.

end.

CLASSES UTEIS

=============

/\* exemplo 1 \*/

USING PROGRESS.json.ObjectModel.JsonObject.

DEF VAR oObj AS JsonObject NO-UNDO.

oObj = NEW JsonObject().

oObj:add('Codigo', 1).

oObj:add('Empresa', 'Totvs').

oObj:add('Data', today).

/\* exemplo 2 \*/

USING PROGRESS.json.ObjectModel.JsonObject.

USING PROGRESS.json.ObjectModel.JsonArray.

DEFINE VARIABLE oObj AS JsonObject NO-UNDO.

DEFINE VARIABLE aList AS JsonArray NO-UNDO.

aList = NEW JsonArray().

for each customer:

oObj = NEW JsonObject().

oObj:add('codigo', customer.CustNum).

oObj:add('nome', customer.Name).

aList:add(oObj).

end.

DEFINE VARIABLE cTmp AS CHARACTER NO-UNDO.

cTmp = STRING(aList:GetJsonText()).

MESSAGE cTmp VIEW-AS ALERT-BOX.

Fazer -> Apos o final do FOR EACH, adicionar um novo JsonObject com o codigo = 44 e o nome = "Catolica", depois disso adicionar no JsonArray

/\* exemplo 3 \*/

DEFINE TEMP-TABLE ttCustomer NO-UNDO LIKE Customer.

FOR EACH Customer NO-LOCK:

CREATE ttCustomer.

BUFFER-COPY Customer TO ttCustomer.

END.

TEMP-TABLE ttCustomer:WRITE-JSON("FILE", "c:/tmp/customer.json", TRUE, ?, ?, TRUE).

EMPTY TEMP-TABLE ttClientes.

Fazer -> Gerar um arquivo JSON com SalesRep

/\* exemplo 4 \*/

USING Progress.Json.ObjectModel.JsonObject.

USING Progress.Json.ObjectModel.JsonArray.

DEFINE VARIABLE oObj AS JsonObject NO-UNDO.

DEFINE VARIABLE aSales AS JsonArray NO-UNDO.

DEFINE VARIABLE aCust AS JsonArray NO-UNDO.

DEFINE VARIABLE iCont AS INTEGER NO-UNDO.

DEFINE VARIABLE iCont2 AS INTEGER NO-UNDO.

aSales = NEW JsonArray().

FOR EACH SalesRep NO-LOCK:

aCust = NEW JsonArray().

iCont = 0.

FOR EACH customer OF SalesRep NO-LOCK:

oObj = NEW JsonObject().

oObj:add('codigo', customer.CustNum).

oObj:add('nome', customer.Name).

aCust:add(oObj).

ASSIGN iCont = iCont + 1.

IF iCont > 3 THEN

LEAVE.

END.

oObj = NEW JsonObject().

oObj:add('salesRep', salesRep.SalesRep).

oObj:add('salesName', salesRep.RepName).

oObj:add('customers', aCust).

aSales:add(oObj).

ASSIGN iCont2 = iCont2 + 1.

IF iCont2 > 3 THEN

LEAVE.

END.

DEFINE VARIABLE cTmp AS CHARACTER NO-UNDO.

cTmp = substr(aSales:GetJsonText(),1,4000).

MESSAGE cTmp VIEW-AS ALERT-BOX.

FUNCOES

=======

/\* exemplo 1 \*/

message "can-find(first Customer)=" can-find(first Customer) skip

"entry(3,'aaa,bbb,ccc,ddd')=" entry(3,"aaa,bbb,ccc,ddd") skip

"Num-entries('banana','a')=" Num-entries("banana","a") skip

'substr("abacaxi", 4, 2)=' substr("abacaxi", 4, 2) skip

'string(121.21,"9999.999")=' string(121.21,"9999.999") skip

'length("teste de escrita")=' length("teste de escrita") skip

'substitute("&1 foi eleito","O presitente")=' substitute("&1 foi eleito","O presitente") skip

'trim(" teste ") + "x" =' trim(" teste ") + "x" skip

'replace("banana", "a", "xyz")=' replace("banana", "a", "xyz") skip

'index("teste de escrita", "de")=' index("teste de escrita", "de")

view-as alert-box.