Exemple vba ave sql

Option Compare Database

Const BobPath As String = "c:\bob50"

Const BobUser As String = "MACRO"

Const BobPassWord As String = "3125"

Const BobCompany As String = "DI"

Sub ImportFactHachette()

'Cette procédure permet l'importation des données de la facture Hachette à partir d'un fichier Excel

'Le fichier est lié à la base de donnée comme une table extérieur qui se trouve dans

'T:\Procédures\Cycle Financier\O\_Organisation de la clôture comptable\Automatisation Liasse\Factures Hachette

'La procédure vérifie que les factures contenues dans le fichier n'ont pas déjà été importées

'Dans ce cas, deux tables sont alimentées :

' - TbHdInvHach qui contient l'entête de la facture (N°, date,...

' - TbLnInvHach qui contient chaque ligne

'Les données ainsi importées sont ensuite traitées pour être implémantées dans les tables du programme comptable Bob

'en vue d'être ensuite validée dans l'application Bob

Dim dbsMe As DAO.Database

Dim rstLstFact As DAO.Recordset

Dim rstHd As DAO.Recordset

Dim rstTot As DAO.Recordset

Dim Annee As String

Dim Run As Integer

Dim NrFact As Long

Dim DtFact As Long

Dim JrFact As Long

Dim MsFact As Long

Dim AnFact As Long

Dim TotFact As Double

Dim stSql1 As String

Dim stSql2 As String

Dim stSql3 As String

Dim stSql4 As String

Dim stSql5 As String

Dim stSql6 As String

Dim stSql7 As String

Set dbsMe = CurrentDb

'Définition de la requête permettant d'obtenir la liste des factures se trouvant dans le fichier à importer (Rq1)

stSql1 = "SELECT Fact\_Hach\_A\_Importer.[N° run ] as Run, " + \_

"Fact\_Hach\_A\_Importer.[N° facture] as NrFact, " + \_

Chr(34) + "20" + Chr(34) + " & Left([Fact\_Hach\_A\_Importer]![Date de facture],2) AS Annee, "

stSql2 = "Fact\_Hach\_A\_Importer.[Date de facture] as DateF, " + \_

"Fact\_Hach\_A\_Importer.[N° client] as NuCli "

stSql3 = "FROM Fact\_Hach\_A\_Importer " + \_

"GROUP BY Fact\_Hach\_A\_Importer.[N° run ], Fact\_Hach\_A\_Importer.[N° facture], " + Chr(34) + "20" + Chr(34) + " & Left([Fact\_Hach\_A\_Importer]![Date de facture],2), " + \_

"Fact\_Hach\_A\_Importer.[Date de facture], Fact\_Hach\_A\_Importer.[N° client];"

Set rstLstFact = dbsMe.OpenRecordset(stSql1 + stSql2 + stSql3)

While Not rstLstFact.EOF

With rstLstFact

Run = .Fields("Run")

Annee = .Fields("Annee")

NrFact = .Fields("NrFact")

DtFact = .Fields("DateF")

End With

stSql1 = "SELECT TbHdInvHach.\* " + \_

"FROM TbHdInvHach " + \_

"WHERE (((TbHdInvHach.Hach\_AnneeFact)=" + Chr(34) + Annee + Chr(34) + ") AND ((TbHdInvHach.Hach\_NrRun)=" + CStr(Run) + ") AND ((TbHdInvHach.Hach\_NrFact)=" + CStr(NrFact) + "));"

Set rstHd = dbsMe.OpenRecordset(stSql1)

If rstHd.EOF Then 'Cette facture n'a pas encore été importéé

'Définition de la requête permettant d'obtenir le total de la facture (Requete8)

stSql1 = "SELECT Fact\_Hach\_A\_Importer.[N° run ], Fact\_Hach\_A\_Importer.[N° facture], Sum(Fact\_Hach\_A\_Importer.[net HT]) AS TotalFact "

stSql2 = "FROM Fact\_Hach\_A\_Importer "

stSql3 = "GROUP BY Fact\_Hach\_A\_Importer.[N° run ], Fact\_Hach\_A\_Importer.[N° facture] "

stSql4 = "HAVING (((Fact\_Hach\_A\_Importer.[N° run ])=" + CStr(Run) + ") AND ((Fact\_Hach\_A\_Importer.[N° facture])=" + CStr(NrFact) + "));"

Set rstTot = dbsMe.OpenRecordset(stSql1 + stSql2 + stSql3 + stSql4)

If Not rstTot.EOF Then

TotFact = rstTot.Fields("TotalFact")

Else

TotFact = 0

End If

Call AAMMJJ(rstLstFact!DateF, AnFact, MsFact, JrFact)

With rstHd

.AddNew

!Hach\_AnneeFact = rstLstFact!Annee

!Hach\_NrRun = rstLstFact!Run

!Hach\_NrFact = rstLstFact!NrFact

!Hach\_NuClient = rstLstFact!nucli

!Hach\_DateFacture = DateSerial(AnFact, MsFact, JrFact)

!StatutBob = "N"

!Hach\_TotFact = TotFact

If TotFact >= 0 Then

!Bob\_Hdbk = "022"

Else

!Bob\_Hdbk = "027"

End If

.Update

End With

'ajouter les lignes dans la table TbLnInvHach (Requete2)

stSql1 = "INSERT INTO TbLnInvHach ( Hach\_AnneeFact, Hach\_NrRun, Hach\_NrFact, Hach\_Nuart, Hach\_Remise, Hach\_PPHTUnit, Hach\_NuEdit, Hach\_Qte, Hach\_NetHtva, Hach\_TVA, Hach\_NetTTC ) "

stSql2 = "SELECT " + Annee + " AS Expr1, Fact\_Hach\_A\_Importer.[N° run ], Fact\_Hach\_A\_Importer.[N° facture], Right(" + Chr(34) + "0000000" + Chr(34) + " & [Fact\_Hach\_A\_Importer]![NUART],7) AS NUART, "

stSql3 = "Max(Fact\_Hach\_A\_Importer.Remise) AS MaxDeRemise, Max(Fact\_Hach\_A\_Importer.[PP HT]) AS [MaxDePP HT], Max(Fact\_Hach\_A\_Importer.NUEDI) AS MaxDeNUEDI, "

stSql4 = "Sum(Fact\_Hach\_A\_Importer.qté) AS Qte, Sum(Fact\_Hach\_A\_Importer.[net HT]) AS CAnet, Sum(Fact\_Hach\_A\_Importer.TVA) AS TVA, "

stSql5 = "Sum(Fact\_Hach\_A\_Importer.[Net TTC]) AS CATvac "

stSql6 = "FROM Fact\_Hach\_A\_Importer " + \_

"GROUP BY " + Annee + ", Fact\_Hach\_A\_Importer.[N° run ], Fact\_Hach\_A\_Importer.[N° facture], Right(" + Chr(34) + "0000000" + Chr(34) + " & [Fact\_Hach\_A\_Importer]![NUART],7) "

stSql7 = "HAVING (((Fact\_Hach\_A\_Importer.[N° run ])=" + CStr(Run) + ") AND ((Fact\_Hach\_A\_Importer.[N° facture])=" + CStr(NrFact) + "));"

DoCmd.RunSQL (stSql1 + stSql2 + stSql3 + stSql4 + stSql5 + stSql6 + stSql7)

End If

Set rstHd = Nothing

rstLstFact.MoveNext

Wend

End Sub

Sub AAMMJJ(DateNum As Long, AA As Long, MM As Long, JJ As Long)

AA = Int(DateNum / 10000)

MM = Int((DateNum - AA \* 10000) / 100)

JJ = DateNum - AA \* 10000 - MM \* 100

End Sub

Sub MajLn()

Dim stSql1 As String

Dim stSql2 As String

Dim stSql3 As String

Dim stSql4 As String

'Mettre à jour les champs de TbLnInvHach à partir de la table D\_Artic (Requete 3)

stSql1 = "UPDATE TbLnInvHach INNER JOIN dbo\_D\_ARTIC\_DWH ON TbLnInvHach.Hach\_Nuart = dbo\_D\_ARTIC\_DWH.COD\_HACHET "

stSql2 = "SET TbLnInvHach.Dili\_Codif = [dbo\_D\_ARTIC\_DWH]![COD\_ART], " + \_

"TbLnInvHach.Dili\_Remise = [dbo\_D\_ARTIC\_DWH]![TAU\_REMFOUR], " + \_

"TbLnInvHach.Dili\_PPHT = [dbo\_D\_ARTIC\_DWH]![PRX\_VTEHTVAFR], "

stSql3 = "TbLnInvHach.COD\_EDITHACH = [dbo\_D\_ARTIC\_DWH]![COD\_EDITHACH], " + \_

"TbLnInvHach.COD\_GROUPE = [dbo\_D\_ARTIC\_DWH]![COD\_GROUPE] "

stSql4 = "WHERE (((TbLnInvHach.Dili\_Codif) Is Null));"

DoCmd.RunSQL (stSql1 + stSql2 + stSql3 + stSql4)

End Sub

Sub TransfertBob()

Dim dbsMe As DAO.Database

Dim rstLstFact As DAO.Recordset

Dim stSql1 As String

Dim stSql2 As String

Dim stSql3 As String

Dim stSql4 As String

Dim stSql5 As String

Set dbsMe = CurrentDb

'Définition de la requête permettant d'obtenir la liste des factures à transférer (Requete5)

stSql1 = "SELECT TbHdInvHach.Hach\_AnneeFact, TbHdInvHach.Hach\_DateFacture, TbHdInvHach.Hach\_NrRun, TbHdInvHach.Hach\_NrFact, " + \_

"TbHdInvHach.Hach\_TotFact, TbHdInvHach.Bob\_Hdbk, TbHdInvHach.Bob\_Hdocno, TbHdInvHach.Bob\_Hyear, " + \_

"TbHdInvHach.Bob\_HFyear, TbHdInvHach.Bob\_Hmonth, TbHdInvHach.StatutBob "

stSql2 = "FROM TbHdInvHach "

stSql3 = "WHERE (((TbHdInvHach.StatutBob) =" + Chr(34) + "N" + Chr(34) + ")) "

stSql4 = "ORDER BY TbHdInvHach.Bob\_Hdbk, TbHdInvHach.Hach\_DateFacture, TbHdInvHach.Hach\_NrRun, TbHdInvHach.Hach\_NrFact;"

Set rstLstFact = dbsMe.OpenRecordset(stSql1 + stSql2 + stSql3 + stSql4)

If Not rstLstFact.EOF Then

Call AjHdDonneeEncodageBob(rstLstFact)

Call AjKhBob(rstLstFact)

Call AjKLBob(rstLstFact)

'Mettre à jour statut bob N -> E

With rstLstFact

.MoveFirst

While Not .EOF

.Edit

!StatutBob = "E"

.Update

.MoveNext

Wend

End With

End If

End Sub

Sub AjHdDonneeEncodageBob(rstHd As DAO.Recordset)

'Cette routine attribue les données d'encodage à chaque record da la tbHdInvHach pour lesquels le champ statutBob = N passé en paramètre

Dim dbsMe As DAO.Database

Set dbsMe = CurrentDb

Dim rstKH As DAO.Recordset

Dim rstAhisto As DAO.Recordset

Dim bTest As Boolean

Dim stSql1 As String

Dim stSql2 As String

Dim stSql3 As String

Dim stSql4 As String

Dim stSql5 As String

Dim Hdbk As String

Dim Mois As Integer

Dim NoDoc As Long

Dim Annee As String

rstHd.MoveFirst

While Not rstHd.EOF

Hdbk = rstHd.Fields("Bob\_Hdbk")

'Vérification s'il existe déjà des écritures dans le KHxxx où xxx = hdbk

stSql1 = "SELECT Max(ac\_kh" + Hdbk + ".TDOCNO) AS TDOCNO, Max(ac\_kh" + Hdbk + ".TFYEAR) AS TFYEAR, Max(ac\_kh" + Hdbk + ".TMONTH) AS TMONTH " + \_

"FROM ac\_kh" + Hdbk + ";"

Set rstKH = dbsMe.OpenRecordset(stSql1)

If Not IsNull(rstKH!TDOCNO) Then 'Il y a des records dans KHxxx

Annee = rstKH.Fields("TFYEAR")

Mois = rstKH.Fields("TMONTH")

NoDoc = rstKH.Fields("TDOCNO") + 1

Else

'Recherche des valeur dans Ahisto (Requete 10)

Annee = rstHd!Hach\_AnneeFact

stSql1 = "SELECT ac\_ahisto.HDBK, ac\_ahisto.HFYEAR, Max(ac\_ahisto.HMONTH) AS LastMonth, Max(ac\_ahisto.HDOCNO) AS LastHDOCNO " + \_

"FROM ac\_ahisto " + \_

"GROUP BY ac\_ahisto.HDBK, ac\_ahisto.HFYEAR " + \_

"HAVING (((ac\_ahisto.HDBK)=" + Chr(34) + Hdbk + Chr(34) + ") AND ((ac\_ahisto.HFYEAR)=" + Chr(34) + Annee + Chr(34) + "));"

Set rstAhisto = dbsMe.OpenRecordset(stSql1)

If Not rstAhisto.EOF Then

Mois = rstAhisto.Fields("LastMonth")

NoDoc = rstAhisto.Fields("LastHDOCNO") + 1

Else

Mois = 1

If Hdbk = "022" Then NoDoc = 220001

If Hdbk = "027" Then NoDoc = 270001

End If

End If

If Month(rstHd.Fields("Hach\_DateFacture")) > Mois Then Mois = Month(rstHd.Fields("Hach\_DateFacture"))

bTest = True

While bTest

rstHd.Edit

rstHd!Bob\_Hdocno = NoDoc

rstHd!Bob\_Hyear = CInt(Annee)

rstHd!Bob\_HFyear = Annee

rstHd!Bob\_Hmonth = Mois

rstHd.Update

NoDoc = NoDoc + 1

rstHd.MoveNext

If Not rstHd.EOF Then

If rstHd.Fields("Bob\_Hdbk") = Hdbk Then

bTest = True

Else

bTest = False

End If

Else

bTest = False

End If

Wend

Wend

End Sub

Sub AjKhBob(rstHd As DAO.Recordset)

'Cette routine complète les tables ac\_KH027 et ac\_KH022 en fonction de la liste de factures encodées transmise dans le recordset rstHD

'Les champs de ce recordset sont :

' - .Hach\_AnneeFact

' - .Hach\_DateFacture

' - .Hach\_NrRun

' - .Hach\_NrFact

' - .Hach\_TotFact

' - .Bob\_Hdbk

' - .Bob\_Hdocno

' - .Bob\_Hyear

' - .Bob\_HFyear

' - .Bob\_Hmonth

' - .StatutBob

Dim bOk As Boolean

Dim bSameHdbk As Boolean

Dim Hdbk As String

Dim DueDate As Date

rstHd.MoveFirst

Set bobkh = New BOBData

bobkh.Path = BobPath

bobkh.Company.User = BobUser

bobkh.Company.Id = BobCompany

bobkh.Company.Password = BobPassWord

bobkh.dataset.DatabaseName = "\\alinfv83\bob50\data\di"

While Not rstHd.EOF

Hdbk = rstHd.Fields("Bob\_Hdbk")

'Ouverture Bob KH en fonction du journal

bobkh.dataset.TableName = "ac\_kh" & Hdbk & ".adt"

bobkh.dataset.Open

bSameHdbk = True

While bSameHdbk

With rstHd

'Calcul de la date d'échéance en fonction de la date facture (90 jours fin de mois = 1/m+4/aa -1)

DueDate = DateSerial(Year(DateAdd("m", 4, rstHd!Hach\_DateFacture)), Month(DateAdd("m", 4, rstHd!Hach\_DateFacture)), 1) - 1

'Ajout du record dans KHxxx

bOk = bobkh.dataset.Append

bOk = bobkh.dataset.Replace("TDBK", !Bob\_Hdbk)

bOk = bobkh.dataset.Replace("TFYEAR", !Hach\_AnneeFact)

bOk = bobkh.dataset.Replace("TYEAR", CInt(!Hach\_AnneeFact))

bOk = bobkh.dataset.Replace("TMONTH", !Bob\_Hmonth)

bOk = bobkh.dataset.Replace("TDOCNO", !Bob\_Hdocno)

bOk = bobkh.dataset.Replace("TDOCDATE", !Hach\_DateFacture)

bOk = bobkh.dataset.Replace("TTYPCIE", "S")

bOk = bobkh.dataset.Replace("TCOMPAN", "F01333")

bOk = bobkh.dataset.Replace("TCIEINFO", "HACHETTE ED. ET DIF.FRANCOPH.")

bOk = bobkh.dataset.Replace("TCIEWAR", False)

bOk = bobkh.dataset.Replace("TCIESEC", False)

bOk = bobkh.dataset.Replace("TDUEDATE", DueDate)

bOk = bobkh.dataset.Replace("TDUECAT", "90F")

bOk = bobkh.dataset.Replace("TDISDATE", !Hach\_DateFacture)

bOk = bobkh.dataset.Replace("TAMOUNT", Abs(Round(!Hach\_TotFact, 2)))

bOk = bobkh.dataset.Replace("TREMINT", Right("000" & !Hach\_NrRun, 3) & "." & Right("00000" & !Hach\_NrFact, 5))

bOk = bobkh.dataset.Replace("TREMEXT", Right("000" & !Hach\_NrRun, 3) & "." & Right("00000" & !Hach\_NrFact, 5))

bOk = bobkh.dataset.Replace("TINTMODE", "S")

bOk = bobkh.dataset.Post

End With

rstHd.MoveNext

If Not rstHd.EOF Then

If rstHd.Fields("Bob\_Hdbk") = Hdbk Then

bSameHdbk = True

Else

bSameHdbk = False

End If

Else

bSameHdbk = False

End If

Wend

bobkh.dataset.Close

Wend

End Sub

Sub AjKLBob(rstHd As DAO.Recordset)

'Cette routine complète les tables ac\_KL027 et ac\_KL022 en fonction de la liste de factures encodées transmise dans le recordset rstHD

'Les champs de ce recordset sont :

' - .Hach\_AnneeFact

' - .Hach\_DateFacture

' - .Hach\_NrRun

' - .Hach\_NrFact

' - .Hach\_TotFact

' - .Bob\_Hdbk

' - .Bob\_Hdocno

' - .Bob\_Hyear

' - .Bob\_HFyear

' - .Bob\_Hmonth

' - .StatutBob

Dim dbsMe As DAO.Database

Dim rstSynthFact As DAO.Recordset

Dim stSql1 As String

Dim stSql2 As String

Dim stSql3 As String

Dim stSql4 As String

Dim stSql5 As String

Dim bOk As Boolean

Dim bSameHdbk As Boolean

Dim Annee As String

Dim Run As Long

Dim NrFact As Long

Dim Hdbk As String

Dim NrLn As Long

Set dbsMe = CurrentDb

rstHd.MoveFirst

'Ouverture Bob KL

Set bobkl = New BOBData

bobkl.Path = BobPath

bobkl.Company.User = BobUser

bobkl.Company.Id = BobCompany

bobkl.Company.Password = BobPassWord

bobkl.dataset.DatabaseName = "\\alinfv83\bob50\data\di"

While Not rstHd.EOF

Hdbk = rstHd.Fields("Bob\_Hdbk")

'Ouverture de ac\_KLxxx en fonction du journal

bobkl.dataset.TableName = "ac\_kl" & Hdbk & ".adt"

bobkl.dataset.Open

bSameHdbk = True

While bSameHdbk

With rstHd

Annee = !Hach\_AnneeFact

Run = !Hach\_NrRun

NrFact = !Hach\_NrFact

End With

'Définition de la requête permettant d'obtenir la synthèse d'imputation de la facture (Requete 4)

stSql1 = "SELECT TbHdInvHach.Hach\_AnneeFact, TbHdInvHach.Hach\_NrRun, TbHdInvHach.Hach\_NrFact, TbHdInvHach.Hach\_DateFacture, " + \_

"TbLnInvHach.COD\_EDITHACH, TbLnInvHach.COD\_GROUPE, Tb\_Numero\_Compte.N°compte, Sum(TbLnInvHach.Hach\_NetHtva) AS Montant "

stSql2 = "FROM (TbHdInvHach INNER JOIN TbLnInvHach ON (TbHdInvHach.Hach\_NrFact = TbLnInvHach.Hach\_NrFact) " + \_

"AND (TbHdInvHach.Hach\_NrRun = TbLnInvHach.Hach\_NrRun) " + \_

"AND (TbHdInvHach.Hach\_AnneeFact = TbLnInvHach.Hach\_AnneeFact)) "

stSql3 = "INNER JOIN Tb\_Numero\_Compte ON (TbLnInvHach.COD\_GROUPE = Tb\_Numero\_Compte.Groupe\_editeur) " + \_

"AND (TbLnInvHach.COD\_EDITHACH = Tb\_Numero\_Compte.Type\_remuneration) "

stSql4 = "GROUP BY TbHdInvHach.Hach\_AnneeFact, TbHdInvHach.Hach\_NrRun, TbHdInvHach.Hach\_NrFact, " + \_

"TbHdInvHach.Hach\_DateFacture, TbLnInvHach.COD\_EDITHACH, TbLnInvHach.COD\_GROUPE, Tb\_Numero\_Compte.N°compte "

stSql5 = "HAVING (((TbHdInvHach.Hach\_AnneeFact)=" + Chr(34) + Annee + Chr(34) + ") " + \_

"AND ((TbHdInvHach.Hach\_NrRun)=" + CStr(Run) + ") " + \_

"AND ((TbHdInvHach.Hach\_NrFact)=" + CStr(NrFact) + "));"

Set rstSynthFact = dbsMe.OpenRecordset(stSql1 + stSql2 + stSql3 + stSql4 + stSql5)

NrLn = 1

While Not rstSynthFact.EOF

'Ajout du record dans KLxxx

bOk = bobkl.dataset.Append

bOk = bobkl.dataset.Replace("TDBK", rstHd!Bob\_Hdbk)

bOk = bobkl.dataset.Replace("TFYEAR", rstHd!Bob\_HFyear)

bOk = bobkl.dataset.Replace("TYEAR", rstHd!Bob\_Hyear)

bOk = bobkl.dataset.Replace("TMONTH", rstHd!Bob\_Hmonth)

bOk = bobkl.dataset.Replace("TDOCNO", rstHd!Bob\_Hdocno)

bOk = bobkl.dataset.Replace("TDOCLINE", NrLn)

bOk = bobkl.dataset.Replace("TTYPELINE", "S")

bOk = bobkl.dataset.Replace("TACCOUNT", rstSynthFact!N°compte)

bOk = bobkl.dataset.Replace("TAMOUNT", Abs(Round(rstSynthFact!Montant, 2)))

bOk = bobkl.dataset.Replace("TWITHBNKVAT", False)

bOk = bobkl.dataset.Replace("TBASVAT", Abs(Round(rstSynthFact!Montant, 2)))

bOk = bobkl.dataset.Replace("TVATTOTAMN", Abs(Round(rstSynthFact!Montant \* 0.06, 2)))

bOk = bobkl.dataset.Replace("TVATAMN", Abs(Round(rstSynthFact!Montant \* 0.06, 2)))

bOk = bobkl.dataset.Replace("TVATDBLAMN", Abs(Round(rstSynthFact!Montant \* 0.06, 2)))

bOk = bobkl.dataset.Replace("TBASLSTAMN", Abs(Round(rstSynthFact!Montant, 2)))

bOk = bobkl.dataset.Replace("TVATDBL", True)

bOk = bobkl.dataset.Replace("TVDEDUC", 100)

bOk = bobkl.dataset.Replace("TVATINST", True)

bOk = bobkl.dataset.Replace("TVATTAX", "Y")

bOk = bobkl.dataset.Replace("TVATBASE", "U")

bOk = bobkl.dataset.Replace("TVATTYPE", "E")

bOk = bobkl.dataset.Replace("TVATNAT1", "MD")

bOk = bobkl.dataset.Replace("TVATNAT2", "HG")

bOk = bobkl.dataset.Replace("TVATCMP", 6)

bOk = bobkl.dataset.Replace("TVSTORED", "ESTD 6")

If rstSynthFact!Montant >= 0 Then

bOk = bobkl.dataset.Replace("TDC", "D")

Else

bOk = bobkl.dataset.Replace("TDC", "C")

End If

bOk = bobkl.dataset.Replace("TREM", "HACHETTE ED. ET DIF.FRANCOPH.")

bOk = bobkl.dataset.Replace("COST\_DEPT", "D100")

bOk = bobkl.dataset.Post

rstSynthFact.MoveNext

NrLn = NrLn + 1

Wend 'Fin encodage de la synthèse du document

rstHd.MoveNext

If Not rstHd.EOF Then

If rstHd.Fields("Bob\_Hdbk") = Hdbk Then

bSameHdbk = True

Else

bSameHdbk = False

End If

Else

bSameHdbk = False

End If

Set rstSynthFact = Nothing

Wend 'Fin de traitement des documents d'un même journal

bobkl.dataset.Close

Wend

End Sub