Excel VBA ADOxを使用して、接続元のAccessを起動するVBAプログラムを記述して下さい。①データを取得する接続元のAccessを起動する。②起動した接続元のAccessから別の接続先のAccessデータベースに対してテーブルデータを取得するための、接続を確立する。③接続先のAccessデータベースから所定のテーブルデータをSQLで再構成する。④再構成したデータを接続元のAccessにテーブルデータとして取り込む。⑤テーブルデータの左端に上段のデータに1から昇順に附番を行う。⑥附番後のテーブルデータをCSVファイルとして所定の保存先に保存する。⑦接続先のAccessを閉じる。

ChatGPT

以下は、Excel VBA ADOxを使用して、接続元のAccessを起動するVBAプログラムです。

vbnet

Copy code

Sub ImportDataFromAccess()

' 変数の宣言

Dim cn As ADODB.Connection

Dim rs As ADODB.Recordset

Dim strSQL As String

Dim strConn As String

Dim appAccess As Access.Application

' Accessアプリケーションを起動する

Set appAccess = New Access.Application

appAccess.Visible = False

appAccess.OpenCurrentDatabase "C:\path\to\your\database.accdb"

' 接続先のデータベースに接続する

Set cn = New ADODB.Connection

strConn = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\path\to\your\other\database.accdb;"

cn.Open strConn

' SQLを使用してデータを再構成する

strSQL = "SELECT \* FROM YourTable WHERE SomeCondition = 'SomeValue'"

Set rs = New ADODB.Recordset

rs.Open strSQL, cn

' データを接続元のAccessに取り込む

appAccess.DoCmd.TransferSpreadsheet \_

TransferType:=acImport, \_

SpreadsheetType:=acSpreadsheetTypeExcel12, \_

TableName:="YourTable", \_

FileName:="C:\path\to\your\output\file.xlsx", \_

HasFieldNames:=True, \_

Range:="YourTable$A1"

' 附番する

appAccess.Run "YourProcedureName"

' 附番後のテーブルをCSVファイルとして保存する

appAccess.DoCmd.TransferText \_

TransferType:=acExportDelim, \_

TableName:="YourTableWithNumbering", \_

FileName:="C:\path\to\your\output\file.csv", \_

HasFieldNames:=True

' 接続先のデータベースを閉じる

rs.Close

cn.Close

' 接続元のAccessを閉じる

appAccess.CloseCurrentDatabase

appAccess.Quit

Set appAccess = Nothing

End Sub

このコードでは、appAccess.OpenCurrentDatabaseを使用して、接続元のAccessをバックグラウンドで起動しています。また、Excelにデータを取り込むために、appAccess.DoCmd.TransferSpreadsheetを使用しています。附番するには、Accessに保存されているプロシージャをappAccess.Runで呼び出しています。最後に、CSVファイルとして保存するためにappAccess.DoCmd.TransferTextを使用しています。

Excel VBA ADOxを使用して、接続元のAccessをバックグラウンドで起動するVBAプログラムを記述して下さい。①バックグラウンドでデータを取得する接続元のAccessを起動する。②起動した接続元のAccessから別の接続先のAccessデータベースに対してテーブルデータを取得するための、接続を確立する。③接続先のAccessデータベースから所定のテーブルデータをSQLで再構成する。④再構成したデータを接続元のAccessにテーブルデータとして取り込む。⑤テーブルデータの左端に上段のデータに1から昇順に附番を行う。⑥附番後のテーブルデータをCSVファイルとして所定の保存先に保存する。⑦接続先のAccessを閉じる。

ChatGPT

以下は、接続元のAccessをバックグラウンドで起動して、別の接続先のAccessデータベースからテーブルデータを取得してCSVファイルに保存するVBAプログラムの例です。接続元と接続先のAccessデータベースのファイルパス、テーブル名、CSVファイルの保存先などは、適宜変更してください。

vbnet

Copy code

Sub ImportDataFromAccess()

' 接続元のAccessデータベースのファイルパス

Dim sourceDBPath As String

sourceDBPath = "C:\path\to\source\Database.accdb"

' 接続先のAccessデータベースのファイルパス

Dim destinationDBPath As String

destinationDBPath = "C:\path\to\destination\Database.accdb"

' テーブル名

Dim tableName As String

tableName = "Table1"

' CSVファイルの保存先

Dim csvFilePath As String

csvFilePath = "C:\path\to\output\file.csv"

' ADOオブジェクトの作成

Dim adoConn As Object

Set adoConn = CreateObject("ADODB.Connection")

' 接続元のAccessデータベースをバックグラウンドで起動

adoConn.Provider = "Microsoft.ACE.OLEDB.12.0"

adoConn.ConnectionString = "Data Source=" & sourceDBPath & ";Jet OLEDB:Database Password=mypassword;"

adoConn.Open

' 接続先のAccessデータベースに接続

Dim sql As String

sql = "SELECT \* FROM " & tableName

Dim rs As Object

Set rs = CreateObject("ADODB.Recordset")

rs.Open sql, "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & destinationDBPath & ";Jet OLEDB:Database Password=mypassword;"

' テーブルデータを取り込む

Dim i As Long

i = 1

Do Until rs.EOF

' 附番

rs.Fields(0).Value = i

' テーブルデータを接続元のAccessに取り込む

Dim insertSql As String

insertSql = "INSERT INTO Table1 (Field1, Field2, Field3) VALUES ('" & rs.Fields(0).Value & "', '" & rs.Fields(1).Value & "', '" & rs.Fields(2).Value & "')"

adoConn.Execute insertSql

rs.MoveNext

i = i + 1

Loop

' CSVファイルに保存する

Dim fs As Object

Set fs = CreateObject("Scripting.FileSystemObject")

Dim ts As Object

Set ts = fs.CreateTextFile(csvFilePath, True, False)

rs.MoveFirst

Do Until rs.EOF

ts.Write rs.Fields(0).Value & "," & rs.Fields(1).Value & "," & rs.Fields(2).Value & vbCrLf

rs.MoveNext

Loop

ts.Close

' 接続を閉じる

rs.Close

Excel VBA ADOxを使用して、接続元のAccessをバックグラウンドで起動するVBAプログラムを記述して下さい。①バックグラウンドでデータを取得する接続元のAccessを起動する。②起動した接続元のAccessから別の接続先のAccessデータベースに対してテーブルデータを取得するための、接続を確立する。③接続先のAccessデータベースから所定のテーブルデータをSQLで再構成する。④再構成したデータを接続元のAccessにテーブルデータとして取り込む。⑤テーブルデータの左端に上段のデータに1から昇順に附番を行う。⑥附番後のテーブルデータをCSVファイルとして所定の保存先に保存する。⑦接続先のAccessを閉じる。

ChatGPT

以下は、Excel VBA ADOxを使用して、接続元のAccessをバックグラウンドで起動するVBAプログラムのサンプルコードです。

vb

Copy code

Sub ImportAccessData()

Dim conn As ADODB.Connection

Dim rs As ADODB.Recordset

Dim strSQL As String

Dim csvFilePath As String

Dim accessFilePath As String

' 接続元のAccessファイルパスとCSV出力ファイルパスを指定する

accessFilePath = "C:\path\to\access\_file.accdb"

csvFilePath = "C:\path\to\output\_file.csv"

' ADOオブジェクトを作成する

Set conn = New ADODB.Connection

Set rs = New ADODB.Recordset

' 接続元のAccessをバックグラウンドで起動する

conn.Open "Provider=Microsoft.ACE.OLEDB.12.0;" & \_

"Data Source=" & accessFilePath & ";" & \_

"Jet OLEDB:Database Password=mypassword;" & \_

"Mode=12;" & \_

"Persist Security Info=False;"

' 接続先のAccessデータベースに接続する

conn.Execute "USE 'C:\path\to\connection\_file.accdb';"

' SQL文を作成する

strSQL = "SELECT \* FROM table1;"

' SQL文を実行して、Recordsetを取得する

Set rs = conn.Execute(strSQL)

' テーブルデータに附番を行う

rs.MoveFirst

Do While Not rs.EOF

rs.Fields(0) = rs.AbsolutePosition

rs.MoveNext

Loop

' CSVファイルにデータを出力する

rs.Save csvFilePath, adPersistCSV

' 接続先のAccessを閉じる

conn.Execute "QUIT;"

' 接続元のAccessを閉じる

conn.Close

' メッセージを表示する

MsgBox "データのインポートが完了しました。"

End Sub

このコードでは、conn.Openメソッドを使用して、バックグラウンドで接続元のAccessを起動しています。その後、conn.Executeメソッドを使用して、接続先のAccessデータベースに接続し、SQL文を実行して、Recordsetを取得しています。そして、取得したデータに附番を行い、CSVファイルに出力しています。最後に、接続先のAccessを閉じて、接続元のAccessを閉じています。