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。ただし、アプリケーション定義またはオブジェクト定義のエラーが発生しないための工夫をしてください。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の上段のデータに1から昇順にAccess上で附番を行う。⑥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

2 / 2

以下は、Excel VBA ADOxを使用して、接続元のAccessを起動し、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、再構成したデータを接続元のAccessに取り込み、テーブルデータに附番し、CSVファイルに保存し、接続先のAccessを閉じるVBAプログラムです。アプリケーション定義またはオブジェクト定義のエラーを避けるために、適切なエラーハンドリングとオブジェクトの解放を行っています。

vb

Copy code

Sub ImportDataFromAccess()

On Error GoTo ErrorHandler

' 接続元のAccessデータベースのパス

Dim sourceDBPath As String

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

' 接続先のAccessデータベースのパス

Dim targetDBPath As String

targetDBPath = "C:\path\to\target\Database.accdb"

' テーブル名

Dim tableName As String

tableName = "TableName"

' CSVファイルの保存先

Dim csvFilePath As String

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

' ADOのオブジェクト変数の宣言

Dim sourceConn As ADODB.Connection

Dim targetConn As ADODB.Connection

Dim targetRs As ADODB.Recordset

Dim cmd As ADODB.Command

' 接続元のAccessデータベースを起動

Set sourceConn = New ADODB.Connection

sourceConn.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & sourceDBPath

' 接続先のAccessデータベースに接続

Set targetConn = New ADODB.Connection

targetConn.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & targetDBPath

' 接続先のAccessデータベースからテーブルデータを取得

Set targetRs = New ADODB.Recordset

targetRs.Open "SELECT \* FROM " & tableName, targetConn

' 再構成したデータを接続元のAccessに取り込む

sourceConn.Execute "DELETE FROM " & tableName

sourceConn.Execute "INSERT INTO " & tableName & " SELECT \* FROM [;" & targetDBPath & "]." & tableName

' テーブルデータに附番

sourceConn.Execute "ALTER TABLE " & tableName & " ADD COLUMN ID AUTOINCREMENT PRIMARY KEY"

sourceConn.Execute "UPDATE " & tableName & " SET ID = (SELECT COUNT(\*) FROM " & tableName & " AS T WHERE T.ID <= " & tableName & ".ID)"

' テーブルデータをCSVファイルに保存

sourceConn.Execute "SELECT \* INTO [Text;HDR=YES;DATABASE=" & csvFilePath & "].[File.csv] FROM " & tableName

' 接続先のAccessを閉じる

targetRs.Close

targetConn.Close

' 接続元のAccessを閉じる

sourceConn.Close

' オブジェクトの解放

Set targetRs = Nothing

Set targetConn = Nothing

Set sourceConn = Nothing

Exit Sub

ErrorHandler:

' エラーハンドリング

MsgBox "エラーが発生しました: " & Err.Description, vbExclamation

On Error Resume Next

' オブジェクトの解放

If Not targetRs Is Nothing Then targetRs.Close

If Not targetConn Is Nothing Then targetConn.Close

If Not sourceConn Is Nothing Then sourceConn.Close

Set targetRs = Nothing

Set targetConn = Nothing

Set sourceConn = Nothing

End Sub

このプログラムでは、エラーハンドリングが追加されており、エラーが発生した場合にエラーメッセージを表示します。また、オブジェクトの解放も適切に行われるようになっています。エラーが発生した場合、オブジェクトを確実に解放してリソースを開放するために、On Error Resume Next としています。

必要に応じてパスやテーブル名を適切に修正し、プログラムを実行してください。エラーメッセージが表示された場合には、エラー内容を確認して対応してください。

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。ただし、アプリケーション定義またはオブジェクト定義のエラーが発生しないための工夫をしてください。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の上段のデータに1から昇順にAccess上で附番を行う。⑥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

3 / 3

以下のコードは、Excel VBA ADOxを使用して接続元のAccessを起動し、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、再構成したデータを接続元のAccessに取り込み、テーブルデータに附番し、CSVファイルに保存し、接続先のAccessを閉じるVBAプログラムです。アプリケーション定義またはオブジェクト定義のエラーが発生しないように、適切なエラーハンドリングを行っています。

vb

Copy code

Sub ImportDataFromAccess()

On Error GoTo ErrorHandler

' 接続元のAccessデータベースのパス

Dim sourceDBPath As String

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

' 接続先のAccessデータベースのパス

Dim targetDBPath As String

targetDBPath = "C:\path\to\target\Database.accdb"

' テーブル名

Dim tableName As String

tableName = "TableName"

' CSVファイルの保存先

Dim csvFilePath As String

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

' ADOのオブジェクト変数の宣言

Dim sourceConn As Object

Dim targetConn As Object

Dim targetRs As Object

Dim cmd As Object

' 接続元のAccessデータベースを起動

Set sourceConn = CreateObject("ADODB.Connection")

sourceConn.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & sourceDBPath

' 接続先のAccessデータベースに接続

Set targetConn = CreateObject("ADODB.Connection")

targetConn.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & targetDBPath

' 接続先のAccessデータベースからテーブルデータを取得

Set targetRs = CreateObject("ADODB.Recordset")

targetRs.Open "SELECT \* FROM " & tableName, targetConn

' 再構成したデータを接続元のAccessに取り込む

sourceConn.Execute "DELETE FROM " & tableName

sourceConn.Execute "INSERT INTO " & tableName & " SELECT \* FROM [;" & targetDBPath & "]." & tableName

' テーブルデータに附番

sourceConn.Execute "ALTER TABLE " & tableName & " ADD COLUMN ID AUTOINCREMENT PRIMARY KEY"

sourceConn.Execute "UPDATE " & tableName & " SET ID = (SELECT COUNT(\*) FROM " & tableName & " AS T WHERE T.ID <= " & tableName & ".ID)"

' テーブルデータをCSVファイルに保存

sourceConn.Execute "SELECT \* INTO [Text;HDR=YES;DATABASE=" & csvFilePath & "].[File.csv] FROM " & tableName

' 接続先のAccessを閉じる

targetRs.Close

targetConn.Close

' 接続元のAccessを閉じる

sourceConn.Close

' オブジェクトの解放

Set sourceConn = Nothing

Set targetConn = Nothing

Set targetRs = Nothing

Set cmd = Nothing

Exit Sub

ErrorHandler:

' エラーメッセージの表示

MsgBox "エラーが発生しました：" & Err.Description, vbExclamation

' 接続先のAccessを閉じる

If Not targetRs Is Nothing Then targetRs.Close

If Not targetConn Is Nothing Then targetConn.Close

' 接続元のAccessを閉じる

If Not sourceConn Is Nothing Then sourceConn.Close

' オブジェクトの解放

Set sourceConn = Nothing

Set targetConn = Nothing

Set targetRs = Nothing

Set cmd = Nothing

End Sub

このコードでは、CreateObjectメソッドを使用してADOのオブジェクトを作成し、エラーハンドリングを追加して例外をキャッチします。エラーが発生した場合には、エラーメッセージを表示し、適切に接続を閉じ、オブジェクトを解放します。

ご使用の環境やデータベースの設定に合わせて、パスやプロバイダのバージョンなどを適宜修正してください。また、必要に応じてエラーハンドリングの部分をカスタマイズすることができます。

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。ただし、アプリケーション定義またはオブジェクト定義のエラーが発生しないための工夫をしてください。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の上段のデータに1から昇順にAccess上で附番を行う。⑥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

4 / 4

以下は、Excel VBA ADOxを使用して、接続元のAccessを起動し、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、再構成したデータを接続元のAccessに取り込み、テーブルデータに附番し、CSVファイルに保存し、接続先のAccessを閉じるVBAプログラムの例です。工夫としては、オブジェクトの作成と解放を明示的に行い、エラーハンドリングを追加しています。

vb

Copy code

Sub ImportDataFromAccess()

On Error GoTo ErrorHandler

' 接続元のAccessデータベースのパス

Dim sourceDBPath As String

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

' 接続先のAccessデータベースのパス

Dim targetDBPath As String

targetDBPath = "C:\path\to\target\Database.accdb"

' テーブル名

Dim tableName As String

tableName = "TableName"

' CSVファイルの保存先

Dim csvFilePath As String

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

' ADOのオブジェクト変数の宣言

Dim sourceConn As ADODB.Connection

Dim targetConn As ADODB.Connection

Dim targetRs As ADODB.Recordset

Dim cmd As ADODB.Command

' 接続元のAccessデータベースを起動

Set sourceConn = New ADODB.Connection

sourceConn.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & sourceDBPath

' 接続先のAccessデータベースに接続

Set targetConn = New ADODB.Connection

targetConn.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & targetDBPath

' 接続先のAccessデータベースからテーブルデータを取得

Set targetRs = New ADODB.Recordset

targetRs.Open "SELECT \* FROM " & tableName, targetConn, adOpenStatic, adLockOptimistic

' 再構成したデータを接続元のAccessに取り込む

sourceConn.Execute "DELETE FROM " & tableName

sourceConn.Execute "INSERT INTO " & tableName & " SELECT \* FROM [;" & targetDBPath & "]." & tableName

' テーブルデータに附番

sourceConn.Execute "ALTER TABLE " & tableName & " ADD COLUMN ID AUTOINCREMENT PRIMARY KEY"

sourceConn.Execute "UPDATE " & tableName & " SET ID = (SELECT COUNT(\*) FROM " & tableName & " AS T WHERE T.ID <= " & tableName & ".ID)"

' テーブルデータをCSVファイルに保存

sourceConn.Execute "SELECT \* INTO [Text;HDR=YES;DATABASE=" & csvFilePath & "].[File.csv] FROM " & tableName

' 接続先のAccessを閉じる

targetRs.Close

targetConn.Close

' 接続元のAccessを閉じる

sourceConn.Close

' オブジェクトの解放

Set targetRs = Nothing

Set targetConn = Nothing

Set sourceConn = Nothing

Exit Sub

ErrorHandler:

MsgBox "エラーが発生しました: " & Err.Description, vbExclamation

On Error Resume Next

' エラー時のクリーンアップ

If Not targetRs Is Nothing Then targetRs.Close

If Not targetConn Is Nothing Then targetConn.Close

If Not sourceConn Is Nothing Then sourceConn.Close

Set targetRs = Nothing

Set targetConn = Nothing

Set sourceConn = Nothing

End Sub

上記のコードは、オブジェクトの作成と解放を明示的に行い、エラーハンドリングも追加しているため、アプリケーション定義またはオブジェクト定義のエラーが発生しづらくなっています。また、エラー発生時にエラーメッセージを表示するためのメッセージボックスも追加しています。

ただし、エラーが発生する具体的な要因によっては、追加の修正が必要になる場合もあります。エラーメッセージやエラーコードがわかる場合は、それを共有していただけると、より具体的なサポートができます。