Excel VBA ADOxを利用して、Base SASを起動して、所定のSASプログラムを実行するVBAの記述をしてください。記述するSASプログラムは、①オラクルデータベースから所定のテーブルデータの取得、②取得したデータをSASに取り込む、③取得したデータをSASでSQL処理する。④SASでSQL処理したデータをCSVファイルとして所定のローカルフォルダに保存する。⑤SASとの接続を切断しVBAを終了する。

ChatGPT

以下は、Excel VBA ADOxを利用して、Base SASを起動して、所定のSASプログラムを実行し、上記の要件を満たすVBAの記述例です。

VB

Copy code

Sub RunSASProgram()

Dim sas As Object

Dim cmd As Object

Dim conn As Object

Dim strSQL As String

Dim sasProgramPath As String

Dim csvFilePath As String

'SASインスタンスを作成

Set sas = CreateObject("SAS.Application")

'SASプログラムを実行するファイルのパスを指定

sasProgramPath = "C:\SAS Programs\sample\_program.sas"

'CSVファイルを保存するローカルフォルダのパスを指定

csvFilePath = "C:\CSV Files\sample\_file.csv"

'SASプログラムを読み込む

sas.DisplayErrors = True

sas.Visible = True

sas.Init

Set cmd = sas.Submit(sasProgramPath)

'ODBC接続を作成し、Oracleデータベースに接続する

Set conn = CreateObject("ADODB.Connection")

conn.Open "Driver={Oracle in OraClient11g\_home1};Dbq=ORACLE\_SID;Uid=username;Pwd=password;"

'取得するテーブルのSQLクエリを作成

strSQL = "SELECT \* FROM sample\_table"

'SQLクエリを実行し、レコードセットを取得

Dim rs As Object

Set rs = CreateObject("ADODB.Recordset")

rs.Open strSQL, conn

'SASにデータを取り込む

cmd.Add "LIBNAME mydata 'C:\SAS Data';"

cmd.Add "PROC IMPORT OUT=mydata.sample\_data"

cmd.Add "DATAFILE='C:\CSV Files\sample\_data.csv'"

cmd.Add "DBMS=CSV REPLACE;"

cmd.Add "GETNAMES=YES;"

cmd.Add "RUN;"

cmd.Submit

'SQL処理を実行するSASプログラムを実行

cmd.Add "%let csv\_file\_path='" & csvFilePath & "';"

cmd.Add "PROC SQL;"

cmd.Add "CONNECT TO ODBC(datasrc=ORACLE\_SID user=username password=password);"

cmd.Add "CREATE TABLE work.sample\_sql\_data AS"

cmd.Add "SELECT \* FROM connection to ODBC("

cmd.Add strSQL

cmd.Add ");"

cmd.Add "DISCONNECT FROM ODBC;"

cmd.Add "QUIT;"

cmd.Submit

'CSVファイルとして保存するSASプログラムを実行

cmd.Add "PROC EXPORT DATA=work.sample\_sql\_data"

cmd.Add "OUTFILE=" & csvFilePath & " REPLACE;"

cmd.Add "RUN;"

cmd.Submit

'SASとの接続を切断し、終了する

rs.Close

conn.Close

cmd.Terminate

Set cmd = Nothing

Set sas = Nothing

End Sub

この例では、ODBC接続を作成してOracleデータベースに接続し、SQLクエリを実行してレコードセットを取得します。

Excel VBA ADOxを利用して、Base SASを起動して、所定のSASプログラムを実行するVBAの記述をしてください。記述するVBAプログラムは、①バックグラウンドでBASE SASを立ち上げる、②SASプログラムを記述した外部のテキストファイルを読み込み、SAS プログラムを実行する。③バックグラウンドのSASとの接続を切断しSASを終了させた後、VBAを終了する。読み込むSASプログラムは以下の内容で記述して下さい。①SASからODBC接続を利用してオラクルデータベースにアクセスする。②オラクルデータベースから所定のテーブルデータを取得する、③取得したデータをSASに取り込む、④取得したデータをSASでSQL処理する。⑤SASでSQL処理したデータをCSVファイルとして所定のローカルフォルダに保存する。

以下は、Excel VBA ADOxを利用して、Base SASを起動して、外部のテキストファイルに記述されたSASプログラムを実行し、バックグラウンドのSASとの接続を切断し、SASを終了させた後、VBAを終了するためのVBAプログラムです。また、読み込むSASプログラムも含めて記述します。

Sub RunSASProgram()

Dim objSAS As Object

Set objSAS = CreateObject("SAS.Application")

' Base SASをバックグラウンドで起動する

objSAS.Visible = False

objSAS.AutomationEcho = False

' SASプログラムを記述したテキストファイルのパスを指定する

Dim sasProgramFilePath As String

sasProgramFilePath = "C:\SASPrograms\sample.sas"

' SASプログラムを読み込んで実行する

Dim sasProgram As String

Dim fso As Object

Set fso = CreateObject("Scripting.FileSystemObject")

Dim ts As Object

Set ts = fso.OpenTextFile(sasProgramFilePath)

sasProgram = ts.ReadAll

ts.Close

' SASからODBC接続を利用してOracleデータベースに接続する

Dim cn As Object

Set cn = CreateObject("ADODB.Connection")

cn.Open "Driver={SAS ODBC Driver};Server=SASApp;Port=0;AuthDomain=SAS Logon Manager;Database=SASUSER;SAS Protocol=IOM;UID=sasdemo;PWD=sasdemo;"

cn.CommandTimeout = 0

' Oracleデータベースから所定のテーブルデータを取得する

Dim rs As Object

Set rs = CreateObject("ADODB.Recordset")

rs.Open "SELECT \* FROM sample\_table", cn

' SASにデータを取り込む

Dim sasData As Object

Set sasData = objSAS.DataEngine.NewDataObject(SAS\_DATA\_TABLE, SAS\_WORK\_LIB)

sasData.CopyFromRecordset rs

' 取り込んだデータをSASでSQL処理する

objSAS.LanguageService.Submit sasProgram

' SASで処理したデータをCSVファイルとして保存する

Dim sasOutput As Object

Set sasOutput = objSAS.CreateObject("SAS.Output")

sasOutput.OutputType = SAS\_OUPTUT\_TYPE\_CSV

sasOutput.OutputObject = "C:\SASOutput\sample\_output.csv"

sasOutput.Export sasData

' SASとの接続を切断し、SASを終了する

objSAS.Quit

End Sub

上記のプログラム中、sasProgramFilePathには、実行するSASプログラムを記述したテキストファイルのパスを指定してください。また、ODBC接続の部分には、接続先のOracleデータベースの情報を設定してください

Excel VBA ADOxを利用して、Base SASをバックグラウンドで起動して、所定のSASプログラムを実行するVBAの記述するとともに、外部から読み込むSASプログラムも分けて記述して下さい。記述するVBAプログラムの手順は、①バックグラウンドでBASE SASを立ち上げる、②SASプログラムを記述した外部のテキストファイルを読み込み、SAS プログラムを実行する。③SASとの接続を切断しSASを終了させる。外部から読み込むSASプログラムは以下の内容で記述して下さい。①SASから所定のオラクルデータベースに対してODBC接続を確立する。②オラクルデータベースから所定のテーブルデータを取得する、③取得したデータをSASに取り込む、④取得したデータをSAS上でSQLにより必要なデータセットに再構成する。⑤再構成したデータセットをCSVファイルとして所定のローカルフォルダに保存する。

以下は、Excel VBA ADOxを利用してBase SASをバックグラウンドで起動して、所定のSASプログラムを実行するVBAプログラムです。また、外部から読み込むSASプログラムも分けて記述しています。なお、本プログラムはWindows環境での動作を想定しています。

<Excel VBAプログラム>

Sub RunSASProgram()

' Base SASをバックグラウンドで起動する

Dim sas As Object

Set sas = CreateObject("SAS.Application")

sas.Visible = False

sas.Wait = True

' 外部からSASプログラムを読み込む

Dim sasProgram As String

sasProgram = ReadSASProgram("C:\Program Files\SAS\SAS 9.4\myprogram.sas")

' SASプログラムを実行する

Dim log As String

log = sas.Submit(sasProgram)

' SASとの接続を切断する

sas.Quit

' 実行結果を出力する

MsgBox log

End Sub

' 外部からSASプログラムを読み込む関数

Function ReadSASProgram(sasProgramPath As String) As String

Dim sasProgram As String

Dim fso As Object

Set fso = CreateObject("Scripting.FileSystemObject")

If fso.FileExists(sasProgramPath) Then

Dim file As Object

Set file = fso.OpenTextFile(sasProgramPath, 1)

sasProgram = file.ReadAll

file.Close

Else

MsgBox "SASプログラムが見つかりませんでした。"

End If

Set fso = Nothing

ReadSASProgram = sasProgram

End Function

以下は、外部から読み込むSASプログラムです。なお、SASからのODBC接続に必要な設定は、事前に行っておく必要があります。

<外部SASプログラム>

/\* SASからオラクルデータベースに接続する \*/

libname mylib odbc dsn='mydsn' uid='myuser' pwd='mypassword';

/\* オラクルデータベースからデータを取得する \*/

data mydata;

set mylib.mytable;

run;

/\* 取得したデータをCSVファイルとして出力する \*/

proc export data=mydata

outfile='C:\Users\myuser\Documents\mydata.csv'

dbms=csv replace;

run;

/\* SASとの接続を切断する \*/

libname mylib clear;

Excel VBA ADOxを利用して、Base SASをバックグラウンドで起動して、所定のSASプログラムを実行するVBAの記述するとともに、外部から読み込むSASプログラムも分けて記述して下さい。記述するVBAプログラムの手順は、①バックグラウンドでBASE SASを立ち上げる、②SASプログラムを記述した外部のテキストファイルを読み込み、SAS プログラムを実行する。③SASとの接続を切断しSASを終了させる。外部から読み込むSASプログラムは以下の内容で記述して下さい。①SASから所定のオラクルデータベースに対してODBC接続を確立する。②オラクルデータベースから所定のテーブルデータを取得する、③取得したデータをSASに取り込む、④取得したデータをSAS上でSQLにより必要なデータセットに再構成する。⑤再構成したデータセットをCSVファイルとして所定のローカルフォルダに保存する。

以下は、Excel VBA ADOxを使用してBase SASをバックグラウンドで起動して、外部のSASプログラムを実行するVBAプログラムです。また、外部から読み込むSASプログラムの内容も記述しています。

外部から読み込むSASプログラムの内容（sas\_program.sas）：

/\* ODBC接続確立 \*/

libname mydb odbc dsn=OracleDB user=myuser password=mypassword;

/\* テーブルデータ取得 \*/

proc sql;

create table work.temp as

select \*

from mydb.mytable;

quit;

/\* データセット再構成 \*/

data work.mydataset;

set work.temp;

where mycondition = 'myvalue';

keep mycolumns;

run;

/\* CSVファイル出力 \*/

proc export data=work.mydataset

outfile='C:\myfolder\mydataset.csv'

dbms=csv replace;

run;

VBAプログラム：

Sub RunSASProgram()

' SASプログラムファイルのパス

Dim sasFilePath As String

sasFilePath = "C:\myfolder\sas\_program.sas"

' Base SASをバックグラウンドで起動する

Dim sas As Object

Set sas = CreateObject("SAS.Application")

sas.Visible = False

' SASプログラムを読み込んで実行する

Dim program As Object

Set program = sas.Submit(sasFilePath)

' プログラムの実行完了を待つ

While program.Status = 0

DoEvents

Wend

' SASとの接続を切断してSASを終了する

sas.Quit

Set sas = Nothing

End Sub

注意点：

ODBC接続に必要なDSN、ユーザー名、パスワードは、実際の値に置き換えてください。

CSVファイルの保存先は、任意のローカルフォルダに変更してください。

バックグラウンドでSASを実行する場合、sas.Visible = Falseと設定する必要があります。