MS Access VBAでパラメータクエリを作成するには、以下の手順を実行します。

クエリデザインを開きます。

パラメータとして使用するフィールド名を含むクエリを作成します。例えば、「SELECT \* FROM Orders WHERE OrderDate Between [Start Date] And [End Date];」のようなクエリを作成します。

クエリデザインを閉じ、VBAエディタを開きます。

パラメータを設定するためのInputBoxを作成します。例えば、「startDate = InputBox("Enter start date:")」のようなコードを作成します。

クエリを実行する前に、パラメータを置き換えます。例えば、「sql = "SELECT \* FROM Orders WHERE OrderDate Between #" & startDate & "# And #" & endDate & "#;"」のようなコードを作成します。

クエリを実行します。例えば、「Set rs = CurrentDb.OpenRecordset(sql)」のようなコードを作成します。

以上の手順によって、MS Access VBAでパラメータクエリを作成することができます。ただし、パラメータの入力値に応じてデータ型をチェックし、適切なエラー処理を実装することが重要です。

MS Access VBAを使用してパラメータクエリを作成するには、次の手順に従ってください。

1. ADO Connectionオブジェクトを作成し、接続先のデータベースに接続します。例えば、以下のようになります。

vbnetCopy code

Dim cnn As New ADODB.Connection cnn.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Users\UserName\Documents\MyDatabase.accdb;" cnn.Open

1. ADO Commandオブジェクトを作成します。このオブジェクトは、SQL文を実行するために使用されます。例えば、以下のようになります。

vbnetCopy code

Dim cmd As New ADODB.Command cmd.ActiveConnection = cnn cmd.CommandText = "SELECT \* FROM Customers WHERE Country=?;"

1. ADO Parameterオブジェクトを作成し、パラメータを設定します。例えば、以下のようになります。

mathematicaCopy code

Dim prm As New ADODB.Parameter prm.Type = adVarChar prm.Size = 50 prm.Direction = adParamInput prm.Value = "USA" cmd.Parameters.Append prm

1. Commandオブジェクトを実行して、レコードセットを取得します。例えば、以下のようになります。

vbnetCopy code

Dim rs As ADODB.Recordset Set rs = cmd.Execute

これで、パラメータを使用したクエリを実行することができます。パラメータの値を変更する場合は、ParameterオブジェクトのValueプロパティを変更し、再度Executeメソッドを呼び出すことができます。

Hiro Hamano

適切なエラー処理として想定されるプログラムはどのように記述しますか？

適切なエラー処理を実装するには、以下のような手順に従ってVBAコードを記述することが一般的です。

パラメータの入力値に応じたデータ型をチェックする。例えば、日付形式の入力値が正しい形式であるかを確認することができます。

パラメータに対して入力が必要である場合には、InputBoxを表示してユーザに入力を促す。

ユーザがキャンセルを選択した場合には、エラーメッセージを表示し、処理を中断する。

エラーが発生した場合には、適切なエラーメッセージを表示し、処理を中断する。

エラーが発生しなかった場合には、クエリを実行し、結果を処理する。

以下は、上記の手順に基づいた例となります。

VBA

Copy code

Dim startDate As Date

Dim endDate As Date

' パラメータの入力値に応じたデータ型をチェックする

On Error Resume Next

startDate = CDate(InputBox("Enter start date:"))

If Err.Number <> 0 Then

MsgBox "Invalid date format. Please enter a valid date.", vbExclamation, "Error"

Exit Sub

End If

On Error GoTo 0

' パラメータに対して入力が必要である場合には、InputBoxを表示する

If startDate = 0 Then

MsgBox "You must enter a start date.", vbExclamation, "Error"

Exit Sub

End If

endDate = CDate(InputBox("Enter end date:"))

If endDate = 0 Then

MsgBox "You must enter an end date.", vbExclamation, "Error"

Exit Sub

End If

' ユーザがキャンセルを選択した場合には、エラーメッセージを表示し、処理を中断する

If startDate = vbNullString Or endDate = vbNullString Then

MsgBox "You must enter both a start date and an end date.", vbExclamation, "Error"

Exit Sub

End If

' エラーが発生した場合には、適切なエラーメッセージを表示し、処理を中断する

On Error GoTo Error\_Handler

Dim sql As String

sql = "SELECT \* FROM Orders WHERE OrderDate Between #" & startDate & "# And #" & endDate & "#;"

Set rs = CurrentDb.OpenRecordset(sql)

' クエリを実行し、結果を処理する

Do While Not rs.EOF

' レコードの処理

rs.MoveNext

Loop

' エラーが発生しなかった場合には、処理を正常終了する

Exit Sub

Error\_Handler:

MsgBox "An error occurred while processing your request. " & Err.Description, vbCritical, "Error

パラメータクエリの作成時に、パラメータの入力値に応じてデータ型をチェックし、適切なエラー処理を実装するためには、以下のようなVBAコードを使用できます。

vbnet

Copy code

'パラメータクエリを作成する

Dim cmd As New ADODB.Command

cmd.ActiveConnection = cnn

cmd.CommandText = "SELECT \* FROM Customers WHERE Country=?;"

'パラメータを作成する

Dim prm As New ADODB.Parameter

prm.Type = adVarChar

prm.Size = 50

prm.Direction = adParamInput

'パラメータの入力値をチェックする

Dim paramValue As Variant

paramValue = InputBox("Enter a country name:")

If IsNull(paramValue) Or Trim(paramValue) = "" Then

MsgBox "Please enter a country name.", vbExclamation, "Error"

Exit Sub

ElseIf Not IsString(paramValue) Then

MsgBox "Invalid input. Please enter a valid country name.", vbExclamation, "Error"

Exit Sub

End If

'パラメータの値を設定する

prm.Value = paramValue

'パラメータをコマンドオブジェクトに追加する

cmd.Parameters.Append prm

'クエリを実行する

Dim rs As ADODB.Recordset

Set rs = cmd.Execute

'レコードセットを処理する

'...

この例では、パラメータの入力値をチェックするために、IsNull、Trim、およびIsStringという関数を使用しています。IsNull関数は、渡された引数がNullであるかどうかをチェックし、Trim関数は、文字列の先頭と末尾のスペースを削除します。IsString関数は、渡された引数が文字列であるかどうかをチェックします。

パラメータの入力値が無効な場合は、適切なエラーメッセージを表示し、サブルーチンから抜け出すようにしています。有効な値が渡された場合は、パラメータの値を設定して、コマンドオブジェクトに追加します。

このように、パラメータの入力値をチェックして適切なエラー処理を実装することで、不正な入力値によるエラーを回避できます。