

Manipulando Strings no VBA (Replace, Mid e InStr)

por Eduardo Vieira Machado (Good Guy)

Hoje gostaria de tratar de um assunto bastante atraente no Access e que é muito trabalhado nas empresas - a manipulação de strings. Entenda manipular como saber trabalhar com os caracteres que a formam, localizar um caracter, substituí-lo por outro, recortar parte de uma string à direita ou à esquerda dela, concatenar strings, etc.

Para isso seria importante conhecer bem as funções do Access que nos possibilitam alcançar este objetivo. A princípio, começaremos com uma função bem conhecida chamada Replace:

(1) Uso da Função Replace:

O Ajuda do Access destaca o seguinte:

Descrição

Retorna uma seqüência de caracteres em que uma subseqüência de caracteres especificada foi substituída por outra subseqüência de caracteres uma quantidade de vezes especificada.

Sintaxe

Replace(*expression*, *find*, *replace*[, *start*[, *count*[, *compare*]]])

A sintaxe da função **Replace** tem os seguintes [argumentos nomeados](#):

Parte	Descrição
<i>Expression</i> (<i>expressão</i>)	Obrigatória. A expressão de seqüência contendo a subseqüência de caracteres a ser substituída.
<i>Find</i> (<i>procura</i>)	Obrigatória. A subseqüência de caracteres que está sendo procurada.
<i>Replace</i> (<i>substitui</i>)	Obrigatória. A subseqüência de caracteres de substituição.
<i>Start</i> (<i>inicia</i>)	Opcional. A posição dentro de <i>expression</i> onde a pesquisa da subseqüência de caracteres deve iniciar. Se

for omitido, será assumido o valor 1.

Count(conta)

Opcional. A quantidade de substituições de subsequências de caracteres a ser efetuada. Se for omitida, o valor padrão será -1, o que significa que todas as substituições possíveis devem ser efetuadas.

Compare(compara)

Opcional. O valor numérico que indica o tipo de comparação a ser usado ao avaliar subsequências de caracteres. Consulte a seção **Configurações** para obter os valores.

Definições

O argumento *compare* pode ter os seguintes valores:

Constante	Valor	Descrição
VbUseCompareOption	-1	Executa uma comparação usando a configuração da instrução Option Compare .
VbBinaryCompare	0	Executa uma comparação binária.
VbTextCompare	1	Executa uma comparação textual.
VbDatabaseCompare	2	Somente Microsoft Access. Efetua uma comparação, com base nas informações existentes no banco de dados.

Valores retornados

Replace retorna os seguintes valores:

Se	Replace retornará
<i>expression</i> tiver comprimento zero	Seqüência de caracteres de comprimento zero ("")
<i>expression</i> for Null	Um erro.

<i>find</i> tiver comprimento zero	Cópia de <i>expression</i> .
<i>replace</i> tiver comprimento zero	Cópia de <i>expression</i> com todas as ocorrências de <i>find</i> removidas.
<i>start</i> > Len(<i>expression</i>)	Seqüência de caracteres de comprimento zero.
<i>count</i> for 0	Cópia de <i>expression</i> .

Comentários

O valor de retorno da função **Replace** é uma seqüência de caracteres, com substituições efetuadas, que inicia na posição especificada por *start* e termina no final da seqüência de caracteres *expression*. Não é uma cópia da seqüência de caracteres original do início ao fim.

Exemplo Prático (1):

Desenvolvi este exemplo para finalidade de estudo. Crie um formulário com uma caixa de texto (Endereco) não acoplada e no ambiente de código, coloque esta função:

Public Function `Substituir()` As String

On Error Resume Next

Dim srua As String

Dim savn As String

Dim setr As String

If Not IsNull(Endereco) Then ‘Endereco é o nome do campo em sua tabela presente no formulário

srua = Replace(Endereco, "Rua", "R") 'Expression(ou nome do campo), Find(procura a string antiga), Replace(Substitui por esta nova string)

Endereco = srua

setr = Replace(Endereco, "Estrada", "Etr") 'Expression(ou nome do campo), Find(procura a string antiga), Replace(Substitui por esta nova string)

Endereco = setr

savn = Replace(Endereco, "Avenida", "Avn") 'Expression(ou nome do campo), Find(procura a string antiga), Replace(Substitui por esta nova string)

Endereco = savn

Substituir = Endereco

Else

Exit Function

End If

End Function

'Na saída do campo Endereco chame a função.

Private Sub Endereco_Exit(Cancel As Integer)

Call Substituir() 'Chama a função na saída do campo endereço

End Sub

Exemplo Prático(2)

Utilizando a função Replace em consultas Atualização:

(a) Consulta simples

No modo SQL de sua consulta digite o seguinte:

```
UPDATE SuaTabela SET Endereco = Replace(Endereco,"Avenida","Avn");
```

Através de uma consulta atualização fica mais fácil de uma vez só alterar todos os registros trocando Avenida por Avn, Estrada por Etr e outros símbolos.

(b) Consulta Atualização múltipla

No modo SQL de sua consulta digite o seguinte:

```
UPDATE SuaTabela SET Endereco = replace(replace(replace(replace(Endereco,"Estrada","Etr"),"Rua","R"),"Avenida","Avn"),"/","-"); (*)
```

(2) Uso da Função Mid:

O Ajuda do Access fornece a seguinte descrição:

Retorna um **Variant (String)** contendo um número especificado de caracteres de uma seqüência.

Sintaxe

Mid(seqüência, início [, comprimento])

A sintaxe da função **Mid** tem os seguintes **argumentos**:

Argumento	Descrição
<i>seqüência</i>	Obrigatório. ">Expressão em seqüência a partir da qual os caracteres são retornados. Se <i>seqüência</i> contiver Null , Null será retornado.
<i>início</i>	Obrigatório. Long . Posição do caractere no argumento <i>seqüência</i> no qual a parte a ser tomada começa. Se <i>início</i> for maior do que o número de caracteres em <i>seqüência</i> , Mid retornará uma seqüência de comprimento nulo ("").
<i>comprimento</i>	Opcional. Variant (Long) . Número de caracteres a serem retornados. Se omitido ou se houver menos caracteres no texto do que o número definido em <i>comprimento</i> (incluindo o caractere em <i>início</i>), todos os caracteres desde a posição <i>início</i> até o final da seqüência serão retornados.

Comentários

Para determinar o número de caracteres em *seqüência*, use a função **Len**.

OBSERVAÇÃO Use a função **MidB** com dados byte contidos em uma seqüência, como em linguagens de conjunto de caracteres de byte-duplo. Em vez de especificar o número de caracteres, os **argumentos** especificam os números de bytes. Para o código de exemplo que usa **MidB**, consulte o segundo exemplo no tópico Exemplo.

Exemplo Prático (1):

Vou manter o exemplo do Ajuda do Access como instrução preliminar, mas com algumas modificações. Em um formulário vazio coloque um botão (cmdFuncao) e no ambiente de código a seguinte a função:

```
Public Function MeuMid()
```

```
Dim MinhaString, PrimeiraPalavra, UltimaPalavra, MeiaPalavra As String
```

```
MinhaString = "Como trabalhar com Strings" ' Crie uma string.
```

PrimeiraPalavra = Mid(MinhaString, 1, 4) ' Returns "Como".

UltimaPalavra = Mid(MinhaString, 20, 7) ' Returns "Strings".

MeiaPalavra = Mid(MinhaString, 5) ' Returns "trabalhar com Strings".

MsgBox "Essa é a string inicial: '" & MinhaString & """

MsgBox "Mid(MinhaString,1,4). A partir do 1º caracter retorna a primeira palavra: '" & PrimeiraPalavra & """

MsgBox "Mid(MinhaString, 20, 7). A partir do 20º caracter retorna a última palavra: '" & UltimaPalavra & """

MsgBox "Mid(MinhaString, 5). A partir do 5º caracter retorna as palavras do meio '" & MeiaPalavra & """

End Function

Private Sub cmdFuncao_Click()

Call MeuMid

End Sub

Exemplo Prático (2):

Você pode utilizar a função Mid para localizar um caracter em uma string e compará-la com uma letra fazendo uso de **Select Case ou If...Then:**

Desenvolvi esta função para este segundo exemplo. Em um formulário vazio coloque uma caixa de texto não acoplada(txtComparacao) e um botão(cmdComparar).

Public Function ComparaCaracter()

Dim i As Integer

Dim Resultado As String

i = InputBox("Digite a posição da letra da palavra:", "Comparação de Caracteres")

Resultado = Mid(txtComparacao, i, 1) 'Pesquisa uma letra a partir da ordem numérica digitada na InputBox

If Resultado = "A" Then

MsgBox "A letra A significa Alfa"

Elseif Resultado = "B" Then

MsgBox "A letra B significa Beta"

Else

MsgBox "Letra desconhecida"

End If

End Function

Private Sub cmdComparar_Click()

Call ComparaCaracter

End Sub

Faça alguns testes para verificar essa nova possibilidade. Estude um programa que criei chamado **Identificador de Numeração de Peças** e crie seu próprio programa para outras finalidades usando a sua criatividade.

Exemplo Prático (3)

Um usuário do fórum solicitou ajudar para um código que se incluísse somente os três primeiros caracteres do nome do produto em uma determinada tabela. O código ficou assim:

```
Private Sub cmdIncluir_Click()
```

```
On Error Resume Next
```

```
Dim a As String
```

```
Dim b As String
```

```
Dim strSQL As String
```

```
a = InputBox("Digite o nome do produto: ")
```

```
b = Mid(b, 1, 3)
```

```
MsgBox b
```

```
strSQL = "INSERT INTO tblTeste(Produto) VALUES('" & b & "')" 'Variáveis do tipo String você usa o apóstrofo mais as aspas duplas para delimita-la.
```

```
DoCmd.RunSQL strSQL
```

```
End Sub
```

(3) Uso da Função InStr:

O Ajuda do Access fornece a seguinte descrição:

Retorna uma **Variant (Long)** que especifica a posição da primeira ocorrência de uma seqüência de caracteres dentro de outra.

Sintaxe

```
InStr([start, ]string1, string2[, compare])
```


A sintaxe da função **InStr** tem os seguintes [argumentos](#):

Parte	Descrição
<i>start</i>	Opcional. Expressão numérica que define a posição inicial de cada pesquisa. Se omitido, a pesquisa iniciará na posição do primeiro caractere. Se <i>start</i> contiver Null , ocorrerá um erro. O argumento <i>start</i> será necessário, se <i>compare</i> for especificado.
<i>string1</i>	Obrigatória. Expressão de seqüência sendo pesquisada.
<i>string2</i>	Obrigatória. Expressão de seqüência de caracteres procurada.
<i>compare</i>	Opcional. Especifica o tipo de comparação de seqüência de caracteres . Se <i>compare</i> for Null, ocorrerá um erro. Se <i>compare</i> for omitido, a configuração Option Compare determinará o tipo de comparação. Especifique um LCID (LocaleID) válido para usar regras específicas da localidade na comparação.

Definições

As configurações do argumento *compare* são as seguintes:

Constante	Valor	Descrição
vbUseCompareOption	-1	Executa uma comparação usando a configuração da instrução Option Compare .
vbBinaryCompare	0	Executa uma comparação binária.
vbTextCompare	1	Executa uma comparação textual.
vbDatabaseCompare	2	Somente Microsoft Access. Efetua uma comparação, com base nas informações existentes no banco de dados.

Valores retornados

Se	A InStr retornará
<i>string1</i> tiver comprimento zero	0
<i>string1</i> for Null	Null
<i>string2</i> tiver comprimento zero	<i>start</i>
<i>string2</i> for Null	Null
<i>string2</i> não for encontrado	0
<i>string2</i> for encontrado dentro de <i>string1</i>	A posição em que a correspondência foi encontrada
<i>start</i> > <i>string2</i>	0

Comentários

A função **InStrB** é usada com os dados de bytes contidos em uma seqüência de caracteres. Em vez de retornar a posição de caractere da primeira ocorrência de uma seqüência de caracteres dentro de outra, **InStrB** retorna a posição de byte.

Exemplo Prático (1):

Desenvolvi a seguinte sub-rotina abaixo para um determinado programa com o fim de deletar a numeração de uma rua a partir de uma vírgula do endereço. O que eu precisava fazer era localizar a vírgula primeiro com a função InStr e depois deletar a string à sua esquerda. Em um formulário vazio, coloque uma caixa de texto não acoplada(ENDERECO).

Public Sub RemoveTexto()

On Error Resume Next

Dim nENDereco As String

Dim strTexto As String

Dim intPos As Integer

nENDereco = ENDereco

intPos = InStr(1, nENDereco, ",") 'Localiza a vírgula na string de endereço a partir do primeiro caractere

strTexto = Trim(Left(nENDereco, intPos)) 'Remove a string à esquerda da vírgula para inclusão de uma nova numeração pela consulta da minha combo cuja fonte de dados é uma consulta dos endereços pelo campo ENDereco de minha tabela

ENDereco = strTexto

End Sub

No evento duplo-clique de sua caixa de texto coloque o seguinte código:

Private Sub ENDereco_DblClick(Cancel As Integer)

If Not IsNull(ENDereco) Then

Call RemoveTexto 'Chama a sub-rotina que remove os caracteres à esquerda da vírgula

MsgBox "Atualize o campo ENDereco com a nova numeração agora !!!", vbInformation, "CONTROLE DE ORDEM DE SERVIÇO"

Else

MsgBox "O campo ENDereco está em branco !!!", vbQuestion, "CONTROLE DE ORDEM DE SERVIÇO"

End If

End Sub

Exemplo Prático(2):

No código abaixo vamos aplicar este conhecimento desta função de uma outra forma agora, localizando o símbolo em um endereço de email, caso os símbolos tal como “@ “(arroba)e “.”, se não estiverem presentes, não será possível reconhecer o email como válido. Com a mesma caixa de texto (Endereco) teste a sub-rotina no evento Antes de Atualizar:

Private Sub Email_BeforeUpdate(Cancel As Integer)

On Error Resume Next

Dim var As String

var = InStr(1, [Email], "@")

If var = 0 Then ' Não é e-mail, já que não contém @. Não aceita sem o arroba é verdade, mas aceita se digitar até aí.

MsgBox "Email inválido. Por favor, digitar um email válido que contenha o símbolo '@' (arroba)!", vbCritical, "Email?"

With Me

.Senha.Enabled = False

.txtConfirmação.Enabled = False

.Lembrete.Enabled = False

End With

Elseif Len(Me!Email) < 10 Then 'Não aceita se for uma string menor que dez caracteres. Resolve o problema anterior.

MsgBox "Email inválido. Por favor, digitar um email completo e válido!", vbCritical, "Email?"

With Me

.Senha.Enabled = False

.txtConfirmação.Enabled = False

.Lembrete.Enabled = False

End With

Elseif InStr(1, Email, "@", 1) < 2 Then ' Não aceita email que no início só tenha dois caracteres ou menos

MsgBox "Email inválido!", vbCritical, "Email?", vbCritical, "Email?"

With Me

.Senha.Enabled = False

.txtConfirmação.Enabled = False

.Lembrete.Enabled = False

End With

Elseif IsNull(Me!Email) Then ' Não aceita o campo em branco

MsgBox "Email inválido. Por favor, não deixe o campo em branco. Digite um email válido!", vbCritical, "Email?"

With Me

.Senha.Enabled = False

.txtConfirmação.Enabled = False

.Lembrete.Enabled = False

End With

Else

MsgBox "Email válido!Aguarde para verificar se já existe este email cadastrado ...", vbInformation, "Email" 'Valida um email que cumpra todos os requisitos acima

With Me

.Senha.Enabled = True

.txtConfirmação.Enabled = True

.Lembrete.Enabled = True

End With

End If

End Sub

Estude o código completo na aplicação Cadastro de Usuários e Login no Sistema neste site <http://www.svcsoftwarehouse.com>

Agradecimentos:

(*) Agradeço ao mestre João Paulo do Máximo Access por esta instrução aprimorada da função Replace em um dos seus exemplos no fórum ExpertAccess

