

Aula 1 – Planilha para Incluir / Excluir / Consultar dados em uma agenda telefônica (Nome, Endereço, Telefone) residindo em Banco de dados Access ou na WEB (usando um banco espelho no meu site, e uma página asp para repassar os comando SQL e devolver a matriz resposta se o comando sql for um select)

A planilha deverá ter as seguintes informações:

	A	B	C	D	E	F	G
1	<b>Nome</b>	<b>Endereco</b>	<b>Telefone</b>				web
2							p:\agenda.mdb
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							

As colunas A, B e C vão receber a partir da linha 3 o resultado da pesquisa feita pelo usuário.

O string a ser procurado será digitado em a2, e o ícone do homem com a lupa será clicado.

Para inserir uma entrada na agenda, os dados serão digitados nas células de A2 a C2, e a figura da agenda deverá ser clicada.

Para excluir uma entrada da agenda, o nome deverá ser digitado em A2 e a borracha deverá ser clicada.

O relógio é um bônus para o usuário: Ele causará o acerto da data e hora do computador, a partir de um relógio externo confiável (erro menor que 1 segundo)

A agenda poderá residir num local da rede, e seu nome e caminho completo será colocado na

célula. G1 (Na figura está em G2, pronta para ser copiada em G1 para testarmos esta opção).

Preferencialmente a agenda deve estar em um site, e nesse caso coloca-se em G1 a palavra WEB.

Nessa opção, pessoas de todos os locais podem se cadastrar usando a planilha, e também consultar os dados colocados por outras pessoas.

O primeiro ícone corresponde ao pedido de Help, e vai funcionar assim: Clica-se nessa figura e recebe-se uma mensagem solicitando que se clique em outra figura, para receber o help a respeito de sua utilização.

Para implementar esse recurso, temos que definir uma variável global que indicará se o usuário solicitou ajuda ou não (Variável global Booleana).

O valor desta variável vai alterar o processamento dos demais cliques (executa a função ou explica como usá-la).

O endereço do banco em G1 será guardado em uma variável global chamada banco.

Também será mostrado como trabalhar com ranges nomeados, para tornar o código independente do local atual dos dados inspecionados na planilha. Se movermos a célula G1 para outro local desta planilha, ou mesmo para outra planilha desta pasta, a rotina não precisará ser alterada.

Esta aula mostra como podemos construir uma planilha obtendo dados digitados por diversas pessoas a qualquer momento, como usar variáveis Globais e ranges nomeados.

Primeiro passo

Pode-se começar o projeto por qualquer rotina, mas nós vamos aqui começar pelas rotinas mais básicas, testá-las e só então passar para a próxima rotina (processo de construção bottom-up).

Escolhemos então fazer o acesso via WEB, já que o maior trabalho é usar o gravador de macros para obter o código necessário à conexão.

Começamos então gravando uma macro para pegar a data e hora da Brasília (que é a mesma do Rio de Janeiro).

Os passos a serem executados são (Excel até o 2003):

Inserir nova planilha

Menu Dados – Importar dados externos – Nova consulta à WEB. Consulta:

[www.ruimedeiros.eti.br/horacerta.asp](http://www.ruimedeiros.eti.br/horacerta.asp)

Informar como célula para resposta a célula a1

O código gerado pelo gravador de macros será:

1. Sheets("Plan1").Select
2. Sheets.Add
3. With ActiveSheet.QueryTables.Add(Connection:= \_
4. "URL;http://www.ruimedeiros.eti.br/horacerta.asp", Destination:=Range("A1"))
- ..... (várias linhas aqui que não mexeremos)
5. .Refresh BackgroundQuery:=False
6. End With

A partir deste código vamos criar 2 rotinas de uso geral: A primeira para criar uma planilha, escondê-la e devolver o objeto que a representa.

A segunda vai obter uma planilha usando a primeira rotina e então criar uma consulta à WEB. Para torná-la genérica, vamos receber a URL como parâmetro. Além disso, no destination:= devemos explicitar que o range está na planilha recém-criada

Além disso vamos criar uma terceira rotina, para excluir a planilha temporária criada para receber o resultados das queries (ela deve jogar na principal a mensagem de erro).

As rotinas codificadas ficam assim:

Sub CriaPlanilha(x)

Set x = Sheets.Add

x.Visible = False

End Sub

Sub DeletaPlanilha(p)

If LCase(p.Cells(1, 1)) = "erro" Then ' Houve erro no query?

Cells(3, 1) = p.Cells(1, 2) ' Devolve na planilha principal

End If

Application.DisplayAlerts = False ' Não deixa vir a mensagem de alerta

p.Delete ' Exclui a planilha

End Sub

```

Sub AcessaWeb(url, x) ' recebe uma url para pesquisar e devolve uma planilha
  CriaPlanilha x          ' Solicita a criação da planilha x
  Set base = x.Cells(1, 1) ' Aonde virá a resposta à query
  With x.QueryTables.Add(Connection:= _
    "URL;http://" & url, Destination:=base) ' troquei as constantes gravadas
    .Name = "nome"
    .FieldNames = True
    .RowNumbers = False
    .FillAdjacentFormulas = False
    .PreserveFormatting = True
    .RefreshOnFileOpen = False
    .BackgroundQuery = True
    .RefreshStyle = xlInsertDeleteCells
    .SavePassword = False
    .SaveData = True
    .AdjustColumnWidth = True
    .RefreshPeriod = 0
    .WebSelectionType = xlAllTables
    .WebFormatting = xlWebFormattingNone
    .WebPreFormattedTextToColumns = True
    .WebConsecutiveDelimitersAsOne = True
    .WebSingleBlockTextImport = False
    .WebDisableDateRecognition = False
    .Refresh BackgroundQuery:=False
  End With
End Sub

```

Agora vamos criar a rotina de Help.

Quando chamada, ela deve ligar uma variável global e informar que se deve clicar no ícone desejado. Se já a variável já estiver ligada, basta desligá-la e avisar isso ao usuário.

Código:

Definição da variável Global (Antes de todas as Sub do módulo)

Dim ajuda As Boolean

Rotina Help:

```

Sub help()
  If ajuda Then
    MsgBox "Pedido de ajuda desativado." & vbCrLf & "Clique de novo para ativar"
    ajuda = False
  Exit Sub
  End If
  ajuda = True
  MsgBox "Clique no ícone desejado para obter ajuda"
End Sub

```

Todas as rotinas que mostrarem o seu help deverão fazer a variável ajuda = false

A próxima rotina é a que acerta o relógio.

Ela vai chamar a rotina de consulta à web (horacerta.asp) já vista, pois gravamos a macro usando esta consulta.

Depois vai atribuir a data e a hora ao sistema, usando as funções especiais Date e Time do VB.

Esta rotina, como as demais, tem possibilidade de mostrar o próprio help.

```
Sub Horacerta()  
  If ajuda Then  
    ajuda = False  
    MsgBox "Apenas clique no relógio para acertar o relógio do seu computador"  
    Exit Sub  
  End If  
  Range("a2:d1000").ClearContents          ' Limpa a consulta anterior  
  AcessaWeb "www.ruimedeiros.eti.br/Horacerta.asp", x  
  Date = Format(x.Cells(1, 1))  
  Time = Format(x.Cells(2, 1), "hh:mm:ss")  
  MsgBox Format(x.Cells(1, 1), "dd/MM/yyyy") & " " & _  
    Format(x.Cells(2, 1), "hh:mm:ss")      ' mostra a data e hora  
  DeletaPlanilha x  
End Sub
```

Este código mostra também como continuar um comando VB em outra linha: Coloca-se após um espaço em branco ao final da linha um sublinhado ( \_ ) e muda-se de linha

A próxima linha continuará (em qualquer coluna) o comando iniciado anteriormente.

Veremos agora como executar um comando SQL sobre um banco de dados.

A linguagem SQL admite comandos para acesso a dados:

SELECT – Faz uma consulta, retornando uma tabela resultante, chamada de recordset.

UPDATE – Altera uma ou mais linhas de uma tabela do banco (não será usado aqui)

INSERT – Insere uma linha em uma tabela do banco

DELETE – Exclui uma ou mais linhas de uma tabela do banco.

A sintaxe desses comandos depende de qual o banco que estamos utilizando, mas existe um conjunto padrão de comandos com a mesma sintaxe, conhecido como SQL ANS.

Se você for utilizar extensamente acesso a banco de dados, procure material na internet usando como argumentos de pesquisa as palavras em maiúsculas acima.

Use esta rotina sempre que desejar executar um comando SQL:

```
Sub SqlAccess(sql, x)  
  Set con = CreateObject("ADODB.Connection")  
  con.Open "Provider=Microsoft.Jet.Oledb.4.0;data Source=" & Banco  
  Set rst = con.Execute(sql)  
  CriaPlanilha x  
  If Left(LCase(Trim(sql)), 6) = "select" Then  
    x.Cells(1, 1).CopyFromRecordset rst      ' devolve a tabela lida na  
  End If                                     ' planilha criada  
  Set rst = Nothing  
  Set con = Nothing  
End Sub
```

Repare que o nome do banco deverá estar na variável global "Banco"

A rotina a seguir é muito simples. Ela vai preparar a url para chamar a rotina já definida anteriormente, que devolverá uma planilha "x" contendo o resultado.

```
Sub SqlWeb(sql, x)
    url = "www.ruimedeiros.eti.br/sql/executa.asp?SQL=" & sql
    AcessaWeb url, x
End Sub
```

Agora uma rotina que só serve para escolher a rotina de acesso ao banco que será utilizada: A da WEB ou a do Banco Access, dependendo do que foi escrito na célula G1 - range("Banco") da planilha.

Com a sua existência, o programador vai chamar sempre uma rotina para emitir um comando SQL, sem se preocupar aonde ele vai atuar.

```
Sub execsql(sql, x)
    Banco = Range("banco")          ' Banco receberá o conteúdo da célula G1
    Range("a3:d1000").ClearContents ' Limpa o resultado da consulta anterior
    If LCase(Banco) = "web" Then    ' Foi solicitada consulta à WEB
        SqlWeb sql, x
    Else
        SqlAccess sql, x           ' Vai acessar o banco
    End If
End Sub
```

Esta rotina não cria uma planilha. Esta será criada pelas rotinas chamadas, e repassada para o programa chamador dela através da variável do tipo WorkSheet "x".

A rotina que comanda a pesquisa e escolhe que linhas serão mostradas é a seguinte (É chamada pelo ícone do Sherlock)

1. Sub pesquisa()
2. If ajuda Then
3. ajuda = False
4. MsgBox "Escreva na célula A2 um string a pesquisar na agenda" & \_
5. vbLf & "Depois clique aqui para pesquisar"
6. Exit Sub
7. End If
8. quem = Trim(Cells(2, 1))
9. Range("b2:c2").ClearContents
10. sql = "select \* from agendademo"
11. execsql sql, x
12. k = 3
13. For i = 1 To 1000
14. If x.Cells(i, 1) = "" Then Exit For
15. linha = x.Cells(i, 1) & x.Cells(i, 2) & x.Cells(i, 3)
16. If InStr(1, linha, quem, vbTextCompare) > 0 Then
17. For j = 1 To 3
18. Cells(k, j) = x.Cells(i, j)
19. Next
20. k = k + 1
21. End If
22. Next
23. DeletaPlanilha x
24. End Sub

Vamos analisar apenas as linhas que são novidade até este momento:

8 – Função trim (apurar) – devolve um string idêntico ao fornecido como parâmetro, mas sem espaços em branco no início e no fim. Então trim(" a b c ") "a b c"

10 – Select \* from agendademo – É o comando SQL que selecionará todos os campos (\*) da tabela chamada "agendademo"

14 - If x.Cells(i, 1) = "" Then Exit For

Esta sintaxe é permitida para o IF (mas não recomendada)

Em vez de escrevermos 3 linhas: "if then"; comando; "end if " podemos escrever apenas uma, desde que haja apenas um comando após o then.

Outra novidade é o exit for – O fluxo de execução pula para após o próximo Next

16 – Função "instr" – Usada extensivamente em programas que tratam textos. Ela devolve a posição do terceiro parâmetro dentro do segundo, após a posição dada pelo primeiro parâmetro. Se não existir esse string dentro do outro, ela devolve 0.

O último operando é opcional. Se não informado, o casamento deve ser perfeito "abc" <> "Abc" Se for vbTextCompare, as maiúsculas não diferenciam os strings.

Agora a função que é chamada pelo ícone da agenda. Ela vai inserir os dados da linha 2 nos 3 campos da tabela do banco

Sub cadastra()

If ajuda Then

ajuda = False

MsgBox "Escreva nas células da linha 2 o nome, o endereço e o telefone a cadastrar" & \_  
vbLf & "Depois clique aqui para pesquisar"

Exit Sub

End If

sql = "insert into agendademo values('" & Cells(2, 1) & "','" & \_  
Cells(2, 2) & "','" & Cells(2, 3) & "''")"

execsql sql, x

DeletaPlanilha x

End Sub

O comando sql fica, após a substituição dos valores das células da linha 2 (suponha X Y e Z):  
"insert into agendademo values('X','Y','Z')"

Como os 3 campos são do tipo TEXT no banco de dados, devem ser escritos entre pliques (aspas simples). Essa sintaxe é uma das permitidas pelo SQL ANS, e como é a mais simples estamos usando ela aqui. Repare que não precisamos saber os nomes dos campos na tabela, mas temos que saber a sua ordem e que são do tipo TEXT.

Por fim, a rotina que exclui (deleta) uma linha na tabela:

Sub Apaga()

If ajuda Then

ajuda = False

MsgBox "Escreva na célula A2 um nome completo para excluir da agenda"

Exit Sub

End If

sql = "delete from agendademo where nome='" & Trim(Cells(2, 1)) & "''"

execsql sql, x

DeletaPlanilha x

End Sub

Precisamos saber o nome do campo que contém o nome da pessoa (chave de acesso).

O módulo completo está no arquivo texto associado à aula 2 no site do curso, ou diretamente, em [www.ruimedeiros.eti.br/vba/AcessaBanco.bas](http://www.ruimedeiros.eti.br/vba/AcessaBanco.bas)