

## MANIPULANDO BANCOS DE DADOS EM ASP

### 1. Acrescentar Registros

Vamos considerar que temos uma base de dados chamada «basedados.mdb» com pelo menos uma tabela chamada «Tabela». Esta tabela é constituída pelos campos «campo1», «campo2» e «campo3».

Este exemplo utiliza utiliza uma *query* SQL para acrescentar o novo registo à tabela.

```
<%
' Para facilitar a escrita do resto do script é
' sempre boa ideia definir algumas constantes
' (mesmo que algumas sejam variáveis por uma
' questão de facilidade)

' INÍCIO DA DEFINIÇÃO DE CONSTANTES

    ' Vamos definir algumas constantes para
    ' a definição de base de dados.
    ' Para podermos utilizar o MapPath temos que
    ' declarar variáveis em vez de constantes.
    ' Daqui para a frente vamos usá-las
    ' como se de constantes se tratassem.
Dim cn
Dim AccessDB

AccessDB = Server.MapPath("./db_scratch.mdb")

cn = "DRIVER={Microsoft Access Driver (*.mdb)};" _
    & "DBQ=" & AccessDB & ";"

%>

<!-- #INCLUDE FILE="adovbs.inc" -->

<%
' FIM DA DEFINIÇÃO DE CONSTANTES
%>

<%
Dim I
Dim SQL
Dim RS ' Recordset

Set RS = Server.CreateObject("ADODB.Recordset")

SQL = "INSERT INTO Tabela(campo1,campo2,campo3)"
SQL = SQL & " VALUES('Maria',12,'25-12-2000')"

RS.Open SQL, cn

' Por fim fecha-se o Recordset e
' liberta-se a memória utilizada
' pelo objecto
```

```

RS.Close
Set RS = Nothing

' Aqui acaba o exemplo.
' Claro que agora poderíamos apresentar
' todos os dados constantes na tabela,
' utilizado o exemplo já aqui apresentado.

```

```
%>
```

## 2. Apagar Registros com SQL

Vamos necessitar de uma base de dados chamada «basedados.mdb» que contém uma tabela chamada «tabela».

Foi utilizado uma query SQL para apagar o registo.

```

<%
' Tal como no exemplo anterior vamos declarar
' todas as constantes (ou variáveis utilizadas como constantes)
' utilizadas ao longo do script (ver exemplo 1 - Inserção de dados)
' INÍCIO DA DEFINIÇÃO DAS CONSTANTES

    Dim cn
    Dim AccessDB

    AccessDB = Server.MapPath("basedados.mdb")

    cn = "DRIVER={Microsoft Access Driver (*.mdb)};" _
        & "DBQ=" & AccessDB & ";"
' FIM DA DEFINIÇÃO DE CONSTANTES

Dim I
Dim idRegistoParaApagar
Dim SQL

Dim rs
Set rs = Server.CreateObject("ADODB.Recordset")

' O id do registo a apagar é passado
' por parâmetro

idRegistoParaApagar = Request.QueryString("id")

' Se o id do registo passado pela Querystring
' não for numérico, colocamos um que não dê erros
' na string SQL. eu vou colocar o id do registo
' a apagar como 0 porque sei que nunca
' existirá um registo com este id, ou seja,
' não se corre o risco de apagar um registo
' existente sem querer.

```

```

If IsNumeric(idRegistoParaApagar) Then
    idRegistoParaApagar = CLng(idRegistoParaApagar)
Else
    idRegistoParaApagar = 0
End If

' Esta query SQL apaga o registo pretendido

SQL = "DELETE FROM Tabela WHERE id LIKE " & idRegistoParaApagar & ";"
rs.Open SQL, cn

' Mostra a mensagem de Registo Apagado

If idRegistoParaApagar <> 0 Then
    Response.Write "O Registo " & idRegistoParaApagar & " foi
Apagado!"
End If

' Finalmente fechamos o recordset e
' libertamos a memória ocupada pelo
' Recordset.

rs.Close
Set rs = Nothing
%>

```

### 3. Correção com SQL

Esta é uma versão do exemplo anterior. Neste exemplo utilizamos uma *query* em SQL.

```

<%
' Tal com vimos nas duas outras partes deste
' exemplo, vamos definir todas as constantes
' utilizadas.
' Recorda que vamos declarar aqui algumas
' variáveis que, no fundo, se irão comportar
' como constantes e por são declaradas
' nesta secção
' INÍCIO DA DEFINIÇÃO DE CONSTANTES

Dim cn
Dim AccessDB = Server.MapPath("basedados.mdb")

' ODBC
cn = "DRIVER={Microsoft Access Driver (*.mdb)};" _
    & "DBQ=" & AccessDB & ";"

Dim I
Dim idRegistoParaCorrigir
Dim SQL

```

```

Dim rs
Dim rsUpdate

Set rs = Server.CreateObject("ADODB.Recordset")
Set rsUpdate = Server.CreateObject("ADODB.Recordset")

' Recebe o id do registo a corrigir.
' Aqui o id é passado por querystring, mas podiamos
' adaptar isto de forma a que fosse uma função e
' o id do registo fosse passado por parâmetro.
idRegistoParaCorrigir = Request.QueryString("id")

' Verifica se id é número, se não for inicializa-o
' a 0, dado que não existe nenhum registo com esse
' id e não provoca erro no SQL.

If IsNumeric(idRegistoParaCorrigir) Then
    idRegistoParaCorrigir = CLng(idRegistoParaCorrigir)
Else
    idRegistoParaCorrigir = CLng(idRegistoParaCorrigir)
End If

' Vamos agora verificar se ele existe na
' tabela. Se existir, efectua a correcção.
' Se não existir apresenta a mensagem da
' não existência.

SQL = "SELECT * FROM Tabela WHERE id LIKE " & idRegistoParaCorrigir & ";"
rs.Open SQL, cn
if Not (rs.EOF and rs.BOF)
    SQL = "UPDATE Tabela SET campo1='aqualquercoisa', "
    SQL = SQL + " campo2=13, campo3='25/12/2000' "
    SQL = SQL + " WHERE id LIKE " & idRegistoParaCorrigir & ";"
    rsUpdate.Open SQL, cn
    Response.Write "O registo id " & idRegistoParaCorrigir & " foi
alterado com sucesso!"
else
    Response.Write "O registo id " & idRegistoParaCorrigir & " não
existe na tabela!"
End if

' Finalmente fechamos os recordset e libertamos
' a memória utilizada pelos objectos utilizando
' a palavra-chave Nothing

rs.Close
rsUpdate.Close
Set rs = Nothing
Set rsUpdate = Nothing
%>

```