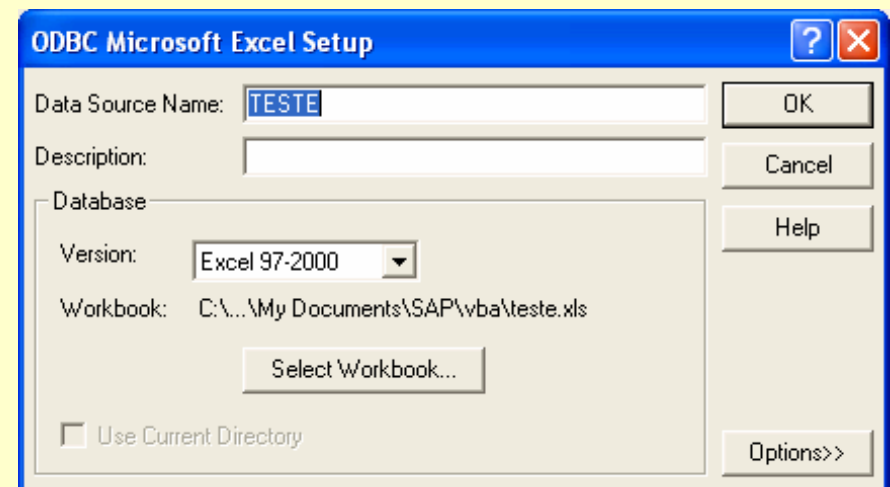
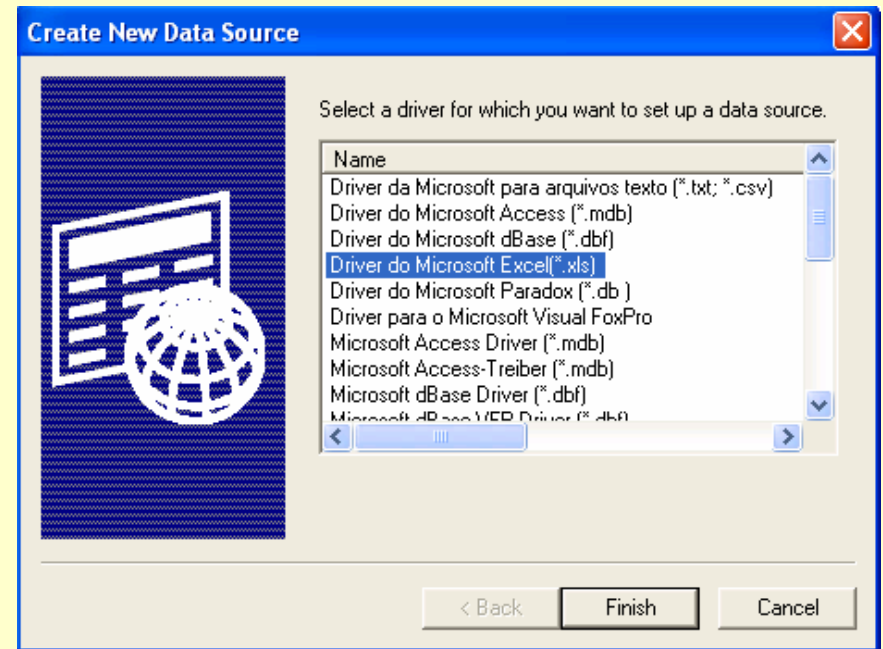
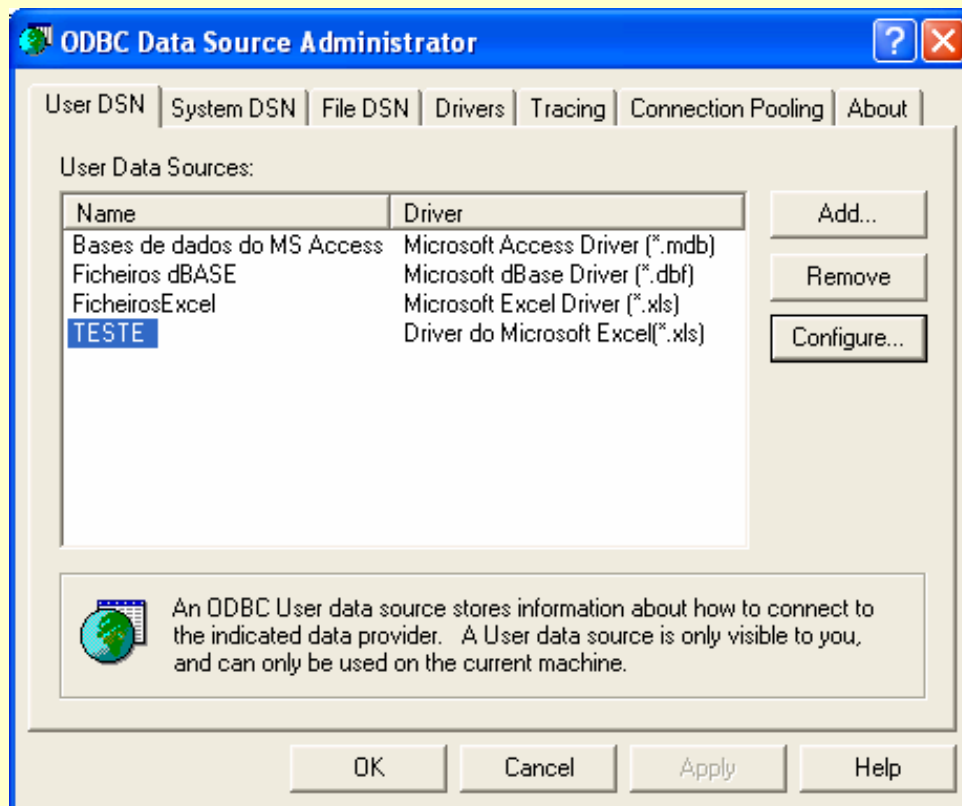


Data Source Names I

● Instalar DSNs

- <Control Panel> seguido de <Administrative Tools> e <Data Sources (ODBC)>



Data Source Names II

● *Exemplo*

```
Dim con As New Connection  
Dim rst As Recordset
```

```
'ligação a livro do Excel  
con.Open "Provider=MSDASQL; DSN=TESTE"
```

```
'criar recordset a partir da ligação DSN  
Set rst = con.Execute("[Folha1$]")
```

```
'para obter previamente o nome das tabelas numa ligação DSN  
Set rst = con.OpenSchema(adSchemaTables)  
'o nome das tabelas (TABLE_NAME) está no campo 2 do conjunto  
'de registos obtidos:  
'          rst.Fields(2).Value  
'para um novo livro do Excel isso resultaria em:  
'          Folha1$ - Folha2$ - Folha3$
```

Objectos Recordset I

■ **Propriedades de ligação**

- **recordset.Source**: fonte de dados do *recordset*
- **recordset.ActiveConnection**: ligação associada ao *recordset*
- **recordset.CursorType**: tipo de cursor do *recordset*
 - **adOpenForwardOnly**: navegação apenas para a frente
 - **adOpenStatic**: navegação livre
 - **adOpenDynamic**: navegação livre; visualização em tempo real das adições e edições (alterações e eliminações) feitas por outros utilizadores
 - **adOpenKeyset**: navegação livre; visualização em tempo real apenas das edições feitas por outros utilizadores; *batch updates*
- **recordset.LockType**: tipo de bloqueio para edição dos registos do *recordset*
 - **adLockReadOnly**: não permite adição nem edição
 - **adLockPessimistic**: bloqueia o registo corrente no início do processo de edição
 - **adLockOptimistic**: bloqueia o registo corrente apenas quando sincroniza os dados com o servidor (método *Update*)
 - **adLockBatchOptimistic**: bloqueia os vários registos alterados apenas quando sincroniza os dados com o servidor (método *UpdateBatch*)

Objectos Recordset II

● *Abrir e Fechar*

- recordset.**Open** [**Source**, **ActiveConnection**, **CursorType**, **LockType**, **Options**]: abre um novo *recordset*
- recordset.**Close**: fecha o *recordset*

● *Movimentação*

- recordset.**Move** **NumRecords**, **Start**: move a posição do registo corrente
- recordset.**MoveFirst**: o mesmo que **.Move 0,adBookmarkFirst**
- recordset.**MoveLast**: o mesmo que **.Move 0,adBookmarkLast**
- recordset.**MoveNext**: o mesmo que **.Move 1,adBookmarkCurrent**
- recordset.**MovePrevious**: o mesmo que **.Move -1,adBookmarkCurrent**
- recordset.**BOF**: indica se a posição do registo corrente é antes do primeiro
- recordset.**EOF**: indica se a posição do registo corrente é depois do último
- recordset.**Bookmark**: marcador que identifica unicamente o registo corrente
- recordset.**RecordCount**: número total de registos

Objectos Recordset III

■ Exemplos

```
Dim rst As New Recordset
```

```
Dim mark As Variant
```

```
Dim cont As Integer
```

```
rst.Open "Clientes", CurrentProject.Connection, adOpenStatic,  
        adLockOptimistic, adCmdTable
```

```
mark = rst.Bookmark 'guarda o marcador do registo corrente
```

```
cont = 0
```

```
rst.MoveFirst
```

```
Do Until rst.EOF
```

```
    If rst.Fields("Cliente") Like "A*" Then cont = cont + 1
```

```
    rst.MoveNext
```

```
Loop
```

```
MsgBox "Existem " & cont & " nomes começados por A em "  
        & rst.RecordCount & " clientes!"
```

```
rst.Bookmark = mark 'repõe o registo guardado como corrente
```

```
rst.Close
```

Objectos Recordset IV

● Edição e adição

- recordset.**AddNew**: para adicionar novos registos
- recordset.**Update**: guarda as alterações feitas ao registo corrente
- recordset.**CancelUpdate**: anula as alterações feitas ao registo corrente

● Exemplos

```
rst.MoveLast
Do Until rst.BOF
    rst.Fields("Taxa de Comissão") = 0.01
    rst.Update
    rst.MovePrevious
Loop
rst.AddNew      'cria um novo registo que passa a ser o corrente
rst.Fields("Cliente") = "António"
resp = MsgBox("Quer adicionar?", vbYesNo)
If resp = vbYes Then
    rst.Update
Else
    rst.CancelUpdate      'o registo corrente volta a ser aquele
                           'que era antes da chamada a AddNew
End If
```

Objectos Recordset V

● *Eliminação*

- recordset.**Delete**: elimina o registo corrente

● *Exemplo*

```
Dim rst As New Recordset
```

```
rst.Open "SELECT * FROM [Clientes] WHERE  
        [Taxa de Comissão]=0", CurrentProject.Connection,  
        adOpenStatic, adLockOptimistic, adCmdText
```

```
rst.MoveFirst
```

```
Do Until rst.EOF
```

```
    rst.Delete
```

```
    rst.MoveNext
```

```
Loop
```

```
rst.Close
```

*'a posição do registo corrente não se altera
'para recolocar válido o registo corrente*

Objectos Recordset VI

● *Batch Updates*

- recordset.**UpdateBatch**: guarda as alterações feitas em todos os registos
- recordset.**CancelBatch**: anula as alterações feitas em todos os registos

● *Exemplo*

```
'para batch updates no Access o cursor precisa de estar do
'lado do cliente (por defeito CursorLocation é adUseServer)
rst.CursorLocation = adUseClient
rst.Open "Clientes", CurrentProject.Connection, adOpenKeyset,
        adLockBatchOptimistic, adCmdTable
rst.AddNew "Cliente", "António"
rst.AddNew "Cliente", "Manuel"
resp = MsgBox("Quer adicionar?", vbYesNo)
If resp = vbYes Then
    rst.UpdateBatch
Else
    rst.CancelBatch      'o registo corrente fica indeterminado
    rst.MoveFirst        'para recolocar válido o registo corrente
End If
rst.Close
```

Objectos Recordset VII

● **Transacções**

- São úteis para quando se pretende garantir o sucesso de uma série de operações que no seu conjunto funcionam como um todo
- Por exemplo, para transferir dinheiro entre duas contas é necessário subtrair uma dada quantidade de um lado e adicionar num outro. Se algumas destas operações falha a outra deixa de fazer sentido e deverá ser anulada
- **connection.BeginTrans**: inicia uma nova transacção
- **connection.CommitTrans**: confirma as alterações feitas durante a transacção e termina a transacção
- **connection.RollbackTrans**: repõe as alterações feitas durante a transacção e termina a transacção
- **connection.Attributes**: atributos para criação automática de transacções
 - **adXactAbortRetaining**: cria uma nova transacção após **RollbackTrans**
 - **adXactCommitRetaining**: cria uma nova transacção após **CommitTrans**

Objectos Recordset VIII

■ Exemplo

```
Dim con As Connection
Dim rst As New Recordset
Dim rst_check As New Recordset

Set con = CurrentProject.Connection
con.Attributes = adXactAbortRetaining + adXactCommitRetaining
con.BeginTrans
rst.Open "Clientes", con, adOpenStatic,
        adLockOptimistic, adCmdTable
rst.AddNew "Cliente", "António"
con.CommitTrans
rst.AddNew "Cliente", "Manuel"
rst_check.Open "Clientes", con, adOpenDynamic,
              adLockOptimistic, adCmdTable
rst_check.MoveLast
MsgBox rst_check.Fields("Cliente")           'apresenta "Manuel"
con.RollbackTrans
MsgBox rst_check.Fields("Cliente")           'dá um erro
```

Objectos Recordset IX

● Filtrar, ordenar e pesquisar registos

- `recordset.Filter`: critério para filtragem dos registos do *recordset*
- `recordset.Sort`: critério para ordenação dos registos do *recordset*
- `recordset.Find Criteria, SkipRows, SearchDirection, Start`: procura o primeiro registo que verifica o critério e torna-o o registo corrente

● Exemplos

```
rst.Filter = "[Taxa de Comissão]=0"
rst.Filter = "" 'elimina qualquer filtro
rst.CursorLocation = adUseClient 'necessário para ordenação
rst.Sort = "[Taxa de Comissão] DESC, Cliente"
rst.Sort = "" 'elimina qualquer ordenação
mark = rst.Bookmark
rst.Find "[Taxa de Comissão]=0", 0, adSearchForward, mark
Do Until rst.EOF 'adSearchBackward como alternativa
    MsgBox rst("Cliente")
    mark = rst.Bookmark
    rst.Find "[Taxa de Comissão]=0", 1, adSearchForward, mark
Loop
```

Tratamento de erros I

● *Ideia básica*

- Sempre que ocorre um erro, o seu tratamento fica a cargo da última declaração **'On Error'** invocada no procedimento corrente
- Caso não haja qualquer declaração desse tipo, o erro é tratado no primeiro procedimento pai que possui tratamento de erros. Se nenhum dos procedimentos envolvidos possuir tratamento de erros, o sistema aborta a execução e apresenta uma mensagem indicando o erro ocorrido

● *Formas de tratar erros*

- **On Error GoTo line**: activa a rotina de tratamento de erros que se encontra a partir da indicação **line**. A posterior ocorrência de qualquer erro transfere imediatamente a execução para a rotina especificada

```
[Sub | Function] procedimento( )  
    On Error GoTo Rotina_Tratamento_Erros  
    [...]  
    Exit [Sub | Function]  
    Rotina_Tratamento_Erros:  
    [...]  
End [Sub | Function]
```

Tratamento de erros II

● Formas de tratar erros

- **On Error Resume Next**: ignora a posterior ocorrência de qualquer erro e prossegue a execução na instrução seguinte à que provocar o erro
- **On Error GoTo 0**: desactiva o tratamento de erros no procedimento corrente

● Retomar a execução

- **Resume**: retoma a execução na instrução que provocou o erro
- **Resume Next**: retoma a execução na instrução seguinte à que provocou o erro
- **Resume line**: retoma a execução na instrução indicada por **line**

● Objecto Err

- **Err.Number**: número que identifica o último erro ocorrido
- **Err.Description**: descrição sumária do último erro ocorrido
- **Err.Clear**: limpa as propriedades **Err.Number** (0) e **Err.Description** (""). É igualmente invocado sempre que uma das seguintes instruções é executada:
 - Instruções do tipo **Resume**
 - Instruções do tipo **On Error**
 - Instruções **Exit Sub**, **Exit Function**, **Exit Property**
- **Err.Raise Number**: provoca a ocorrência do erro identificado por **Number**

Tratamento de erros III

Exemplo

```
Sub teste_erros()  
    On Error GoTo trat_erros:  
    Dim erro As Integer : erro = 5 : MsgBox "Início teste_erros"  
    gera_erro (erro)  
    MsgBox "Fim teste_erros" : Exit Sub  
trat_erros:  
    MsgBox "Erro n. " & Err.Number  
    Select Case Err.Number  
        Case 5 To 6  
            erro = erro + 1 : Resume  
        Case Else  
            Resume Next  
    End Select  
End Sub  
  
Sub gera_erro(erro As Integer)  
    MsgBox "Início gera_erro"  
    Err.Raise erro  
    MsgBox "Fim gera_erro"  
End Sub
```

Sequência de mensagens

- Início teste_erros
- Início gera_erro
- Erro n. 5
- Início gera_erro
- Erro n. 6
- Início gera_erro
- Erro n. 7
- Fim teste_erros