

ESEMPI RECORDSET

Esempio 1) Utilizzo di un Recordset per creare un funzione che riproduce una qualsiasi Dfunction

Codice scritto in un modulo

Enum DF

 LookUp = 1

 Count = 2

 Sum = 3

 Avg = 4

End Enum

```
Public Function My_DFunction(Field As String, Table As String, _  
    Optional Condition As Variant, Optional Kind As DF = LookUp) As Variant
```

```
Dim Rc As Recordset2
```

```
Dim MySQL As String
```

```
On Error GoTo Err:
```

```
    MySQL = "SELECT " & Field & " FROM " & Table
```

```
    If Not IsMissing(Condition) Then MySQL = MySQL & " WHERE " & Condition
```

```
    'Debug.Print MySQL
```

```
    Set Rc = CurrentDb.OpenRecordset(MySQL)
```

```
    If Rc.EOF Then
```

```
        My_DFunction = Null
```

```
    Else
```

```
        Select Case Kind
```

```
            Case LookUp: My_DFunction = Rc.Fields(0)
```

```
            Case Count
```

```
                Rc.MoveLast
```

```
                My_DFunction = Rc.RecordCount
```

```
            Case Else
```

```
                My_DFunction = 0
```

```
                Do While Not Rc.EOF
```

```
My_DFunction = My_DFunction + Rc.Fields(0)
```

```
Rc.MoveNext
```

```
Loop
```

```
End Select
```

```
End If
```

```
If Kind = Avg Then My_DFunction = My_DFunction / Rc.RecordCount
```

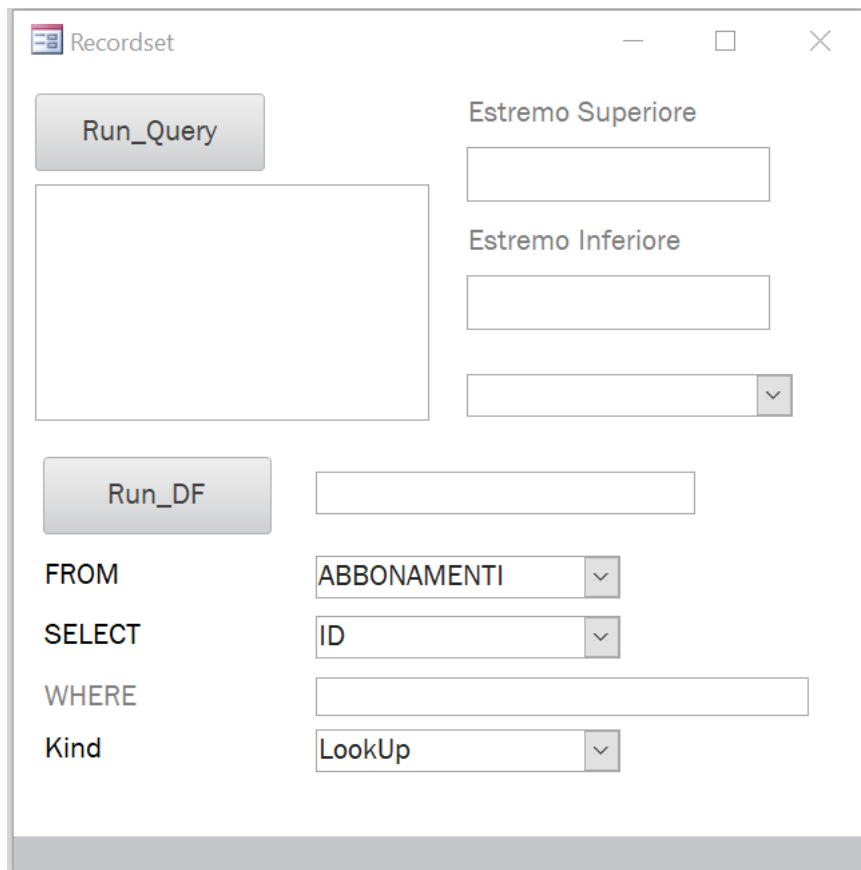
```
Set Rc = Nothing
```

```
Err:
```

```
If Err.Number <> 0 Then My_DFunction = "Errore, parametri errati"
```

```
End Function
```

Esempio 2) Creazione di una maschera che esegue delle query



The screenshot shows a window titled "Recordset" with the following elements:

- A button labeled "Run_Query" at the top left.
- A large empty rectangular area below the "Run_Query" button.
- Two input fields on the right side, labeled "Estremo Superiore" and "Estremo Inferiore".
- A button labeled "Run_DF" below the "Run_Query" button.
- A text label "FROM" followed by a dropdown menu showing "ABBONAMENTI".
- A text label "SELECT" followed by a dropdown menu showing "ID".
- A text label "WHERE" followed by an empty input field.
- A text label "Kind" followed by a dropdown menu showing "LookUp".

Cliccando il pulsante Run Query viene letto il valore scritto in estremo superiore ed in estremo inferiore e viene eseguita una query che restituisce nome e cognome (concatenati) ed età, di tutti gli utenti con età compresa nell'intervallo di input. Tali valori sono usati per popolare la casella di riepilogo e la casella combinata posta sotto al pulsante Run Query.

Cliccando Run_DF si esegue una DFunction (quella selezionata nell'apposito menù a scelta multipla posto in fondo alla maschera). La Dfunction opera sul campo e sulla tabella selezionate applicando, se presente la condizione WHERE scritta nell'apposita casella di testa. Il risultato viene mostrato nella casella di testo (non editabile) a fianco del pulsante.

Si noti che la lista delle tabelle è automaticamente compilata con tutte le tabelle del database. Analogamente la lista dei campi è automaticamente aggiornata con tutti i campi della tabella selezionata nel menù sovrastante.

Il codice è il seguente

```
Private Sub Cmd_RQ_DblClick(Cancel As Integer)
Dim EI As Integer, ES As Integer
Dim MySQL As String
Dim W_Cond As String
Dim Rcs As Recordset2

    EI = CInt(Nz(Me.Txt_EI.Value))
    ES = CInt(Nz(Me.Txt_ES.Value))
    MySQL = "SELECT Concatena(Nome, Cognome)AS F_Name, Age([Data Di Nascita]) As Età FROM
UTENTI"
    W_Cond = "WHERE Age([Data Di Nascita]) BETWEEN " & EI & " AND " & ES
    MySQL = MySQL & " " & W_Cond & " Order By Age([Data Di Nascita]) Desc"
    Debug.Print My_SQL
    Set Rcs = CurrentDb.OpenRecordset(MySQL, dbOpenDynaset)
    Do While Not Rcs.EOF
        'Debug.Print Rcs.Fields(1)
        Rcs.MoveNext
    Loop
    If Rcs.RecordCount > 0 Then
        Me.Cbx_Query.RowSourceType = "Table/Query"
        Me.Cbx_Query.ColumnCount = 2
        Me.Cbx_Query.ColumnWidths = "3cm;2cm"
        Set Me.Cbx_Query.Recordset = Rcs
        Me.Ccb_Query.RowSourceType = "Table/Query"
        Set Me.Ccb_Query.Recordset = Rcs
    Else
        Me.Cbx_Query.RowSourceType = "Value List"
        Me.Cbx_Query.RowSource = "Nobody found"
        Me.Ccb_Query.RowSourceType = "Value List"
        Me.Ccb_Query.AddItem "Nobody found", 0
    End If
End Sub
```

```
Me.Ccb_Query.Value = Me.Ccb_Query.ItemData(0)
Set Rcs = Nothing
End Sub
```

```
Private Sub Comando11_DblClick(Cancel As Integer)
    Me.Txt_Result = My_DF()
End Sub
```

```
Private Sub Form_Load()
Dim Tb As TableDef
Dim Nm As String
'Leggo il nome di tutte le tabelle del Database
For Each Tb In CurrentDb.TableDefs
    Nm = Tb.Name
    If Left(Nm, 3) <> "MSy" And Left(Nm, 3) <> "Tbl" Then Me.Cmb_FROM.AddItem (Tb.Name)
Next Tb
Me.Cmb_FROM.Value = Me.Cmb_FROM.ItemData(0)
Me.Cmb_Kind.Value = Me.Cmb_Kind.ItemData(0)
Me.Txt_Where.Value = ""
'Call Remove_All
Call Populate
End Sub
```

```
Private Sub Cmb_From_AfterUpdate()
    Call Populate
End Sub
```

```
Private Sub Populate()
Dim Tb As TableDef
Dim F As Field
Call Remove_All
For Each Tb In CurrentDb.TableDefs
    If Tb.Name = Me.Cmb_FROM.Value Then
        For Each F In Tb.Fields
            'Debug.Print F.Name
            Me.Cmb_Select.AddItem (F.Name)
        Next F
    End If
Next Tb
Me.Cmb_Select.Value = Me.Cmb_Select.ItemData(0)
End Sub
```

```
Private Function Remove_All()
```

```
On Error GoTo Err
```

```
    Do While Me.Cmb_Select.ListCount > 0
```

```
        Me.Cmb_Select.RemoveItem (0)
```

```
    Loop
```

```
Err:
```

```
End Function
```

```
Public Function My_DF() As Variant
```

```
Dim Rc As Recordset2
```

```
Dim MySQL As String, Field As String, Table As String, Cond As String
```

```
On Error GoTo Err:
```

```
    Field = Me.Cmb_Select.Value
```

```
    Table = Me.Cmb_FROM.Value
```

```
    Cond = Me.Txt_Where.Value
```

```
    Kind = Me.Cmb_Kind.Value
```

```
    MySQL = "SELECT " & Field & " FROM " & Table
```

```
    If Cond <> "" Then MySQL = MySQL & " WHERE " & Cond
```

```
    'Debug.Print MySQL
```

```
    Set Rc = CurrentDb.OpenRecordset(MySQL)
```

```
    If Rc.EOF Then
```

```
        My_DF = Null
```

```
    Else
```

```
        Select Case Kind
```

```
            Case "LookUp": My_DF = Rc.Fields(Field)
```

```
            Case "Count"
```

```
                Rc.MoveLast
```

```
                My_DF = Rc.RecordCount
```

```
            Case Else
```

```
                My_DF = 0
```

```
                Do While Not Rc.EOF
```

```
                    My_DF = My_DF + Rc.Fields(Field)
```

```
                    Rc.MoveNext
```

```
                Loop
```

```
            End Select
```

```
        End If
```

```
        If Kind = "Avg" Then My_DF = My_DF / Rc.RecordCount
```

```
        Set Rc = Nothing
```

```
Err:
```

```
    If Err.Number <> 0 Then My_DF = "Errore, parametri errati"
```

```
End Function
```

Le due funzioni pubbliche usate nella query sono le seguenti.

```
Public Function Age(D As Date) As Integer
    Age = DateDiff("yyyy", D, Date)
    If Month(D) > Month(Date) Then Age = Age - 1
    If Month(D) = Month(Date) Then
        If Day(D) >= Day(Date) Then Age = Age - 1
    End If
End Function
```

```
Public Function Concatena(N As String, C As String) As String
    Concatena = UCase(Left(N, 1)) & "."
    Concatena = Concatena & UCase(Left(C, 1))
    Concatena = Concatena & LCase(Mid(C, 2))
End Function
```

Esempio 3) Ricerca, filtro e ordinamento su recordset

Quattro funzioni equivalenti che usano differenti approcci per trovare l'ID di un utente noto il suo username e la sua password

Scritte nel modulo Rcs_Find_Filter

Funzione basata su metodo Find

```
Public Function Get_ID(User As String, PW As String) As Integer
Dim My_SQL As String
Dim My_Cond As String
Dim Rc As Recordset2
    My_SQL = "SELECT ID, USERNAME, PASSWORD FROM UTENTI"
    My_Cond = "USERNAME = '" & User & "' AND PASSWORD = '" & PW & "'"
    Set Rc = CurrentDb.OpenRecordset(My_SQL)
    Rc.FindFirst (My_Cond)
    If Rc.NoMatch Then
        Get_ID = 0
    Else
        Get_ID = Rc.Fields(0)
    End If
End Function
```

```

    Rc.FindNext (My_Cond)
    If Not Rc.NoMatch Then Get_ID = -1
End If
Set Rc = Nothing
End Function

```

‘Funzione che sfrutta una query già filtrata per aprire il recordset

```

Public Function Get_ID2(User As String, PW As String) As Integer
Dim My_SQL As String
Dim My_Cond As String
Dim Rc As Recordset2
My_Cond = "USERNAME = '" & User & "' AND PASSWORD = '" & PW & "'"
My_SQL = "SELECT ID, USERNAME, PASSWORD FROM UTENTI WHERE " & My_Cond
Set Rc = CurrentDb.OpenRecordset(My_SQL)
If Rc.EOF Then
    Get_ID2 = 0
Else
    Rc.MoveLast
    Get_ID2 = Rc.Fields(0)
    If Rc.RecordCount > 1 Then Get_ID2 = -1
End If
Set Rc = Nothing
End Function

```

‘Funzione che sfrutta DFunction

```

Public Function Get_ID3(User As String, PW As String) As Integer
Dim My_Cond As String
Dim N As Integer
My_Cond = "USERNAME = '" & User & "' AND PASSWORD = '" & PW & "'"
N = Nz(DCount("ID", "UTENTI", My_Cond), 0)
Select Case N
    Case 0: Get_ID3 = 0
    Case 1: Get_ID3 = DLookup("ID", "UTENTI", My_Cond)
    Case Else: Get_ID3 = -1
End Select
End Function

```

‘Funzione che sfrutta il metodo Seek. In questo caso si sfrutta il fatto che i campi username e password della tabella utente sono usati come indice di nome UN_PW

```
Public Function Get_ID4(User As String, PW As String) As Integer
Dim db As Database
Dim Tb As TableDef
Dim Ix As Index
Dim Rc As Recordset2
    Set db = CurrentDb
    Set Tb = db.TableDefs("UTENTI")
    'Each table has an index called Primary Key, in this table there is also an index based on two
fields
    ' This code shows the name of the available indexes
    For Each Ix In Tb.Indexes
        'Debug.Print Ix.Name
    Next Ix
    Set Rc = db.OpenRecordset("UTENTI", dbOpenTable)
    With Rc
        .Index = "UN_PW"
        .Seek "=", User, PW
    End With
    If Not Rc.NoMatch Then Get_ID4 = Rc.Fields("ID")
End Function
```

‘Procedura che mostra il concetto di bookmark

```
Public Sub Bookm(Initial As String)
' examples of recordset movement using bookmarks
Dim WhCond As String
Dim rst As Recordset
Dim bk As String
    Set rst = CurrentDb.OpenRecordset("UTENTI", dbOpenDynaset)
' Bookmarks
    WhCond = "Cognome like '" & Initial & "*'"
    rst.FindFirst WhCond
    Debug.Print "Recod ID is " & rst!ID
    bk = rst.Bookmark ' Place the bookmark
    rst.MoveLast
    Debug.Print "Record ID is " & rst!ID
    rst.Bookmark = bk ' Return to the bookmark
    Debug.Print "Record ID is " & rst!ID & " using the bookmark"
```



```
rst.Close
Set rst = Nothing
End Sub
```

‘Procedura che mostra come filtrare e ordinare un recordset

```
Public Sub Filter_Sort(Optional Ag As Integer = 25, Optional ID As Integer = 40)
Dim Rc1 As Recordset2, Rc2 As Recordset2
Dim MySQL As String
    MySQL = "SELECT ID, Age([Data Di Nascita]) As AGE FROM UTENTI WHERE Age([Data Di Nascita])
<= " & Ag
    Set Rc1 = CurrentDb.OpenRecordset(MySQL)
    Debug.Print Show_All(Rc1)
    Rc1.Filter = "ID >= " & ID
    Rc1.Sort = "AGE DESC"
    Rc1.MoveFirst
    Debug.Print "Sorted?"
    Debug.Print Show_All(Rc1)
    Set Rc2 = Rc1.OpenRecordset
    Debug.Print "Sorted?"
    Debug.Print Show_All(Rc2)
    Set Rc1 = Nothing
    Set Rc2 = Nothing
End Sub
```

```
Private Function Show_All(Rc As Recordset2) As String
Dim F As Field
Dim S As String
    Do While Not Rc.EOF
        For Each F In Rc.Fields
            S = S & CStr(F) & ";"
        Next F
        Rc.MoveNext
        S = S & vbNewLine
    Loop
    Show_All = S
End Function
```