

## Variabili Temporanee

A partire dalla versione 2007, Access ha introdotto la collection **TempVars**, oggetto che semplifica lo scambio di dati fra procedure, specialmente se relative a differenti form. Tale collection viene automaticamente creata dall'applicazione ed è subito pronta per l'uso, senza bisogno di crearla né con un'istruzione Dim né, successivamente, con un'istruzione Set. Attenzione che l'indicizzazione parte da zero.

Tale collection serve a contenere delle variabili, di fatto globali che possono essere gestite mediante le seguenti proprietà e i seguenti metodi:

- TempVars.Add(Nome\_variabile, Valore)
- TempVars(Item), con Item sia posizionale sia stringa
- TempVars.Remove(Nome\_variabile)
- TempVars.RemoveAll
- TempVars.Count

Un semplice esempio è mostrato di seguito:

```
TempVars.Add "Nome", "Francesco" 'Modo standard di aggiungere una variabile
TempVars("Cognome").Value = "Rossi" 'Si può fare anche così
TempVars.Add "Numero", 1 'I tipi possono anche essere differenti, è una collection come le altre...
Debug.Print TempVars!Cognome 'Scriverà Rossi
Debug.Print TempVars.Item(1) 'Scriverà Rossi
Debug.Print TempVars.Item("Cognome") 'Scriverà, ancora, Rossi
Debug.Print TempVars.Item(2) 'Scriverà 1
For Each V In TempVars 'Si può ciclare sulle TempVars
    Debug.Print V.Name
    Debug.Print V.Value
    If V.Value = 1 Then TempVars.Remove ("Numero") 'Cancella la variabile Numero
Next V
```

La collection TempVars è globale e pubblica e, pertanto, tutte le variabili contenute in essa sono direttamente visibili da qualsiasi modulo e/o applicazione. Tali variabili possono essere quindi usate in sostituzione ai metodi standard di passaggio dati tra maschere ossia: (i) le variabili globali e (ii) il parametro open arg (di tipo stringa) del metodo DoCmd.OpenForm.

Un altro vantaggio offerto dalle TempVars, forse il maggiore, è che tali variabili possono essere direttamente richiamate all'interno di query salvate.

Consideriamo un esempio. Supponiamo di avere una tabella corrispondente ad un'anagrafica clienti e di voler creare e salvare una query che restituisca i dati relativi ad un particolare cliente. Tale query sarà qualcosa del genere:

```
SELECT Nome, Cognome, [...]
FROM Anagrafica
WHERE ID = [Inserire ID]
```

Formulata così, si tratta però di una query parametrica e, pertanto, quando eseguita, l'utilizzatore dovrà compilare una maschera d'inserimento in cui specificare il valore dell'ID da cercare.

Per ovviare a questo inconveniente sarebbe bello modificare la clausola WHERE nel modo seguente:

```
WHERE ID = I
```

con I una variabile valorizzata (a run time) tramite codice VBA.

Anche qualora I dovesse essere una variabile globale, questa query non funzionerebbe. Una query, può, infatti, ricevere solo funzioni pubbliche, non variabili pubbliche.

Per ovviare a questo fatto dovremmo allora scrivere (in un modulo pubblico) una banale funzione di tipo Get che legge e restituisce la variabile pubblica V.

Per esempio:

```
Public Function Get_ID() As Integer
    Get_ID = I 'Legge e restituisce la variabile pubblica I
End Function
```

Fatto questo, la query potrebbe essere modificata nel modo seguente:

```
SELECT Nome, Cognome, [...]
FROM Anagrafica
WHERE ID = Get_ID()
```

La soluzione è semplice, ma tediosa, per ogni variabile globale servirebbe, infatti una specifica funzione di tipo Get.

Le variabili temporanee risolvono tale problema perché, come detto, a differenza di quelle globali possono essere direttamente richiamate in un query. Supponiamo allora di aver creato e valorizzato, ad esempio al valore 1, la variabile temporanea di nome "ID" con l'istruzione TempVars("ID",1), in questo caso la query può essere riformulata nel modo seguente:

```
SELECT Nome, Cognome, [...]
FROM Anagrafica
WHERE ID = TempVars("ID")
```

Concludiamo questa breve nota ricordando che anche i valori scritti nei component grafici delle maschere possono essere direttamente richiamati in una query. Supponiamo ad esempio di aver posizionato una Combo Box di nome Cbx\_Id su una maschera di nome Mask e supponiamo inoltre che la Combo Box sia popolata con tutti gli ID dell'anagrafica clienti. In questo caso la query precedente potrebbe anche essere scritta come:

```
SELECT Nome, Cognome, [...]
FROM Anagrafica
WHERE ID = [Maschere]![Mask]![Cbx_Id]
```