Bigino Moduli Access

Utilizzare Tabelle Senza Indici

- ✓ Dichiarare una variabile oggetto di tipo recordset
- ✓ Abbinarla ad un nome di tabella mediante il metodo Open
- ✓ Navigare tra i record o effettuare modifiche

```
Dim rstMiaTabella as Recordset

Set rstMiaTabella = Currentdb.OpenRecordset("nomeTabella", dbOpenDynaset)

'Se non è necessario effettuare modifiche sui record aprire in modalità dbOpenSnapshot
'N.B. in questo caso nomeTabella può essere una Tabella o una query Aggiornabile

rstMiaTabella.MoveFirst ([MoveLast],[MoveNext],[MovePrevious])    'navigazione

if rstMiaTabella.EOF Then … 'Sono alla fine
if rstMiaTabella.BOF Then … 'Sono all'inizio

if rstMiaTabella.EOF and rstMiaTabella.BOF then … 'La tabella è vuota

rstMiaTabella.Edit  'Inizia l'editazione del record corrente
rstMiaTabella.AddNew      'Aggiunge un nuovo record e si prepara per ricevere gli input
rstMiaTabella.Update      'Salva le modifiche o l'inserimento

rstMiaTabella.FindFirst "criterio"      'Trova un record secondo "criterio"
                                        '"criterio" è come la parte WHERE di una query
                                        'senza la parola WHERE es.
                                        ' "NomeCliente ='Andrea'" con apicini
                                        ' "NumeroCliente=125"     senza apicini

if rstMiaTabella.Nomatch then …  'Non trovato il record ricercato

rstMiaTabella.Close              'Chiude la tabella
set rstMiaTabella = Nothing             'Libera memoria dal riferimento all'oggetto
```

Utilizzare Tabelle Con Indici

- ✓ Dichiarare una variabile oggetto di tipo recordset
- ✓ Abbinarla ad un nome di tabella mediante il metodo Open
- ✓ Selezionare l'indice sul quale si desidera lavorare
- ✓ Navigare tra i record o effettuare modifiche

```
Dim rstMiaTabella as Recordset

Set rstMiaTabella = Currentdb.OpenRecordset("nomeTabella", dbOpenTable)

'N.B. in questo caso nomeTabella può essere SOLO una Tabella interna al database
'non si possono aprire in questo modo le tabelle collegate

rstMiaTabella.Index = "nomeIndice"      'Imposta l'indice corrente
                                        'una volta che la tabella è aperta posso usare
                                        'questa istruzione dovunque prima di una
                                        'istruzione Seek
                                        'L'indice attivo può essere solo uno

rstMiaTabella.MoveFirst ([MoveLast],[MoveNext],[MovePrevious])    'navigazione

if rstMiaTabella.EOF Then … 'Sono alla fine
if rstMiaTabella.BOF Then … 'Sono all'inizio

if rstMiaTabella.EOF and rstMiaTabella.BOF then … 'La tabella è vuota

rstMiaTabella.Edit  'Inizia l'editazione del record corrente
rstMiaTabella.AddNew      'Aggiunge un nuovo record e si prepara per ricevere gli
input
rstMiaTabella.Update      'Salva le modifiche o l'inserimento


rstMiaTabella.Seek "operatore",Campo1,Campo2,…, Campo13
                                        'Effettua una ricerca basata sull'indice
                                        'selezionato precedentemente.
                                        ' "operatore" può essere :
                                        ' "="        : ricerca l'occorrenza esatta
                                        ' "<>"       : ricerca il primo diverso
                                        ' ">"        : ricerca il primo maggiore
                                        ' ">="       :      ricerca      il      primo
maggiore/uguale
                                        ' "<"        : ricerca il primo minore
                                        ' "<="       : ricerca il primo minore/uguale

   'N.b. la sequenza dei campi di confronto deve rispettare la sequenza dei campi
   nell' indice: se l'indice ha due campi (il primo di testo ed il secondo numerico)
   allora Campo1 deve essere di testo e Campo2 deve essere numerico.
   Esempio : rstMiaTabella.Seek "=", strNomeCliente, lngNumeroCliente
   Dove   strNomeCliente indica una variabile di tipo stringa
          LngNumeroCliente indica una variabile di tipo long


if rstMiaTabella.Nomatch then …  'Non trovato il record ricercato
```

```
rstMiaTabella.Close              'Chiude la tabella
set rstMiaTabella = Nothing            'Libera memoria dal riferimento all'oggetto
```

Eseguire Query di Access

Si intendono le query di comando ovvero quelle query che non selezionano record ma effettuano solo cancellazioni, aggiornamenti o inserimenti. In caso diverso (query di select) utilizzare gli esempi relativi alle tabelle senza indici.

Se la query è già memorizzata nelle query di Access

Metodo 1: Docmd.OpenQuery "*nomeQuery*"

Metodo 2:

```
Dim qdfMiaQuery as QueryDef

Set qdfMiaQuery = Currentdb.QueryDefs("nomeQuery")
QdfMiaQuery.Execute

Set qdfMiaQuery = Nothing
```

Se la query non è memorizzata in Access ma viene costruita come stringa SQL

Metodo 1:

```
dim sSql as String

sSql = "DELETE * FROM MIATABELLA;"

Currentdb.Execute sSQL
```

Metodo 2:

```
dim sSql as String

sSql = "DELETE * FROM MIATABELLA;"

DoCmd.RunSQL sSql
```

Eseguire Query di Aggiornamento / Inserimento / Cancellazione

Se la query è già memorizzata in Access o la si crea come stringa SQL seguire le istruzioni riportate agli esempi precedenti

Per creare come stringa SQL una query di Aggiornamento la sintassi è la seguente:

```
Dim sSQL as string

SSQL = "UPDATE MITABELLA SET NOMECLIENTE='ANDREA' WHERE NUMEROCLIENTE=19;"

'Oppure

SSQL = "UPDATE MITABELLA SET NOMECLIENTE='ANDREA', INDIRIZZO = 'VIA FRANZAROLA'
WHERE NUMEROCLIENTE=19;"


'Eseguire la query
```

Per creare come stringa SQL una query di Inserimento la sintassi è la seguente:

```
Dim sSQL as string

SSQL = "INSERT INTO MIATABELLA (CAMPO1, CAMPO2, …, CAMPOn) SELECT
ALTRATABELLA.CAMPO1, ALTRATABELLA.CAMPO2, … ALTRATABELLA.CAMPOn FROM ALTRATABELLA
WHERE ALTRATABELLA.CAMPO1 = 'ANDREA';"

'N.b la sequenza dei campi di destinazione deve essere uguale alla sequenza dei
campi di select: se MIATABELLA.CAMPO1 è di testo allora anche ALTRATABELLA.CAMPO1
deve essere di testo. Stesso discorso per CAMPO2,… , CAMPOn
Se i campi di destinazione sono 10 allora anche i campi di select devono essere 10
dello stesso tipo nella medesima sequenza.

'N.b Se MIATABELLA e ALTRATABELLA hanno la medesima struttura e si vogliono
copiare tutti o parte dei record di ALTRATABELLA in MIATABELLA allora per brevità
è possibile scrivere:

SSQL = "INSERT INTO MIATABELLA.* SELECT ALTRATABELLA.* FROM ALTRATABELLA WHERE
ALTRATABELLA.CAMPO1 = …;"

'Eseguire la query
```

Per creare come stringa SQL una query di Eliminazione la sintassi è la seguente:

```
Dim sSQL as string
SSQL = "DELETE * FROM MITABELLA WHERE NUMEROCLIENTE=19 OR NOMECLIENTE='ANDREA';"
'Oppure, se si vuole svuotare completamente la tabella
SSQL = "DELETE * FROM MITABELLA;"
'Eseguire la query
```

Utilizzo di Query con Parametri

Caso 1: la query è di comando (non ritorna record)

```
Dim qdfMiaQuery as QueryDef

Set qdfMiaQuery = Currentdb.QueryDefs("nomeQuery")

qdfMiaQuery.Parameters("nomeParametro1") = …  'Assegnare un valore omogeneo al
qdfMiaQuery.Parameters("nomeParametro2") = …  'tipo dati del parametro
qdfMiaQuery.Parameters("nomeParametro3") = …
…
qdfMiaQuery.Parameters("nomeParametroN") = …

'Eseguire la query
qdfMiaQuery.Execute

Set qdfMiaQuery = Nothing
```

Caso 2: la query ritorna record

```
Dim qdfMiaQuery as QueryDef
Dim rstMiaQuery as Recordset

Set qdfMiaQuery = Currentdb.QueryDefs("nomeQuery")

qdfMiaQuery.Parameters("nomeParametro1") = …  'Assegnare un valore omogeneo al
qdfMiaQuery.Parameters("nomeParametro2") = …  'tipo dati del parametro
qdfMiaQuery.Parameters("nomeParametro3") = …
…
qdfMiaQuery.Parameters("nomeParametroN") = …

'Aprire il recordset risultante dall'esecuzione della query

Set rstMiaQuery = qdfMiaQuery.OpenRecordset


RstMiaQuery.Close
Set rstMiaQuery = Nothing
Set qdfMiaQuery = Nothing
```

Comandi per l'eliminazione di un oggetto

Se l'oggetto è una tabella il comando più breve è il seguente:

```
Currentdb.Execute "DROP TABLE miaTabella;"
'Oppure
Docmd.RunSql "DROP TABLE miaTabella;"
```

Attenzione ! Questi due metodi non controllano se la tabella esiste prima di cancellarla, quindi se la tabella non esiste ritornano un errore.

Per l'elimiazione di ogni tipo di oggetto Access

```
    DoCmd.DeleteObject acForm, "objectname"          'Cancella un form
    DoCmd.DeleteObject acMacro, "objectname"         'Cancella una Macro
    DoCmd.DeleteObject acModule, "objectname"        'Cancella un Modulo
    DoCmd.DeleteObject acQuery, "objectname"         'Cancella una Query
    DoCmd.DeleteObject acReport, "objectname"        'Cancella un report
    DoCmd.DeleteObject acTable, "objectname"         'Cancella una Tabella
```

Attenzione ! Questi due metodi non controllano se l'oggetto esiste prima di cancellarlo, quindi se non esiste ritornano un errore.

Comandi per la creazione/eliminazione di Indici su tabella

Per creare un nuovo indice :

Metodo 1:

```
      Dim tdfMiaTabella as TableDef    'Variabile per la struttura della tabella
      Dim mioIndice As Index           'Variabile per la struttura dell'indice

      'Prepara la struttura
      Set tdfMiaTabella = Currentdb.Tabledefs("NomeTabella")

      'Crea un nuovo indice
      Set mioIndice = tdfMiaTabella.CreateIndex("nomeDellIndice")

      'Associa I campi al nuovo indice
      MioIndice.Fields.Append MioIndice.CreateField("Campo1")
      MioIndice.Fields.Append MioIndice.CreateField("Campo2")
      MioIndice.Fields.Append MioIndice.CreateField("Campo3")
      …
      MioIndice.Fields.Append MioIndice.CreateField("Campo13")

      'Opzionalmente posso specificare se l'indice è Primari o Univoco
      MioIndice.Primary = True
      MioIndice.Unique = True
```

```
'Salva l'indice nella struttura della tabella
tdfMiaTabella.Indexes.Append MioIndice

N.B. Campo1, Campo2 … CampoN sono I nomi dei campi già contenuti in tabella
```

Metodo 2:

il secondo metodo prevede la costruzione di una stringa sql che generi l'indice.
La sintassi è la seguente

```
Dim sSQL as String

'Per un indice primario
SSQL = "CREATE UNIQUE INDEX NOMEINDICE ON NOMETABELLA (CAMPO1, CAMPO2, …, CAMPON)
WITH PRIMARY"

'Per un indice non primario ma univoco
SSQL = "CREATE UNIQUE INDEX NOMEINDICE ON NOMETABELLA (CAMPO1, CAMPO2, …, CAMPON)
;"

'Per un indice non primario non univoco
SSQL = "CREATE INDEX NOMEINDICE ON NOMETABELLA (CAMPO1, CAMPO2, …, CAMPON) ;"


'Eseguire il comando SQL con
Currentdb.execute sSQL
Oppure
Docmd.RunSQL sSQL
```

Per cancellare un indice:

Metodo 1:

```
Dim tdfMiaTabella as TableDef    'Variabile per la struttura della tabella

'Prepara la struttura
Set tdfMiaTabella = Currentdb.Tabledefs("NomeTabella")

'Elimina l'indice indice
tdfMiaTabella.Indexes("nomeDellIndice").Delete
```

Metodo 2:

il secondo metodo prevede la costruzione di una stringa sql che elimini l'indice.
La sintassi è la seguente

```
Dim sSQL as String

SSQL = "ALTER TABLE MIATABELLA  DROP INDEX MIOINDICE;"
```

Aprire Maschere con parametri di collegamento all'oggetto chiamante

Immaginiamo che la maschera che stiamo per aprire sia abbinata ad un recordset all'interno del quale vogliamo visualizzare solo i record in cui il nome corrisponda al nome visualizzato in un'altra maschera (che funziona da chiamante)

Caso 1: Siamo sicuri che se appare la maschera chiamata allora la chiamante deve esistere ed è caricata ed è sempre la stessa. In questo caso progetteremo la maschera chiamata in modo che il suo recordset sia già filtrato dai parametri mostrati nella chiamante.

Caso 2: La maschera chiamata potrebbe essere aperta da diverse maschere che quindi non hanno gli stessi criteri di filtro da passare. In questo caso ogni chiamante passerà alla chiamata la propria stringa per il criterio di selezione:

```
dim sSql as String

sSQL = "[NOME CLIENTE] = '" & me.NomeCliente & "'"
```

n.b. In questo esempio [NOME CLIENTE] è il nome del campo all'interno del recordset della maschera chiamata e me.NomeCliente è il valore del recordset nella chiamante.

```
'Apro ora la maschera chiamata indicandole di leggere solo i record che corrispondono ai criteri passati

DoCmd.OpenForm "nomeMaschera", acNormal, , sSQl
```

N.B. Se la maschera chiamata è basata su una query con parametri non è possibile per il chiamante valorizzare questi parametri prima dell'apertura della maschera chiamata, a meno che i parametri non siano riferimenti a form già caricati ed attivi in precedenza.