

# GAMBAS E SQLITE

di Fea Sergio  
f.surfing@tiscali.it

Questo piccolo articolo vuole essere una semplice guida all' utilizzo base di gambas e sqlite3, rivolto a chi effettua i primi passi con questo linguaggio di programmazione ed ha necessità di realizzare un applicazione stand-alone che debba immagazzinare una certa quantità di dati in db..

## Creazione di un database

Il primo passo che dovremo realizzare sarà appunto creare un database adatto alle nostre esigenze, per far ciò utilizziamo la classe **connection** del componente **gb.db**.

Questa classe ha bisogno di alcuni parametri che andremo a fornirgli in questo modo

```
DIM sName AS String ' sName è una variabile che indica il nome che diamo al nostro database
DIM sPercorso as String ' indica il percorso in cui andremo a creare il db
DIM $hConn as NEW Connection

sName="prova"
sPercorso=user.home 'definiamo come percorso la nostra home

WITH $hConn
.Type = "sqlite3" ' indicazione del tipo di database da utilizzare
.Host = sPercorso
.Login = ""
.Password = ""
END WITH
```

a questo punto possiamo procedere alla creazione verificando però che il db non sia già presente altrimenti visualizzeremo una finestra di errore:

```
$hConn.Open

IF NOT $hConn.Databases.Exist(sName) THEN 'verifica che il db non esista
    $hConn.Databases.Add(sName)          'crea il nostro db
ELSE
    Message.Error("database già esistente", "OK")
ENDIF

$hConn.Close
```

Così abbiamo creato il nostro db ma non ha ancora nessuna tabella all'interno e quindi è inutilizzabile, per realizzare la struttura possiamo utilizzare la funzione `.tablefields.add` oppure sfruttare un query SQL.

Considerando la seconda ipotesi e volendo creare una tabella di nome "biblio" con dei dati biografici quali:

nome  
cognome  
indirizzo

procederemo con una query di questo tipo:

```
sql = "CREATE TABLE 'biblio' ( 'id' INTEGER PRIMARY KEY AUTOINCREMENTAL, 'nome' VARCHAR(255) DEFAULT NULL, 'cognome' VARCHAR(255) DEFAULT NULL, 'indirizzo' VARCHAR(255) DEFAULT NULL);"
```

il tutto scritto in un'unica riga.

Si noti il campo "id" che contiene l'indice del database ed è di tipo integer autoincrementante (ovvero ogni volta che inseriamo un record il rispettivo id verrà inserito automaticamente). Tutti gli altri campi io per abitudine li creo di tipo testo con lunghezza 255 caratteri.

Quindi il nostro codice per la creazione del db sarà così modificato

```
PUBLIC SUB Button1_Click()

DIM sName AS String ' sName è una variabile che indica il nome che diamo al nostro database
DIM sPercorso AS String ' indica il percorso in cui andremo a creare il db
DIM $hConn AS NEW Connection
DIM sql AS String
DIM htable AS Result
sName = "prova"
sPercorso = User.Home 'definiamo come percorso la nostra home

WITH $hConn
.Type = "sqlite3" ' indicazione del tipo di database da utilizzare
.Host = sPercorso
.Login = ""
.Password = ""
END WITH

$hConn.Open

IF NOT $hConn.Databases.Exist(sName) THEN 'verifica che il db non esista
    $hConn.Databases.Add(sName) 'crea il nostro db
    $hConn.Close 'chiude il db
```

```

$hConn.Name = sName
$hConn.Open()                'riapre il db per l' aggiunta della tabella

    sql = "CREATE TABLE 'biblio' ( 'id' INTEGER PRIMARY KEY, 'nome' VARCHAR(255)
DEFAULT NULL, 'cognome' VARCHAR(255) DEFAULT NULL, 'indirizzo' VARCHAR(255)
DEFAULT NULL);"
'scrivere la query su un unica riga oppure separarla con i separatori &

    hTable = $hConn.EXEC(sql)

    $hConn.Commit
    Label1.Text = "Database creato"
ELSE
    Message.Error("database già esistente", "OK")

ENDIF
$hConn.Close

END

```

### Aggiunta di un record

Ora che abbiamo un bel db nuovo fiammante dobbiamo inserire i dati all' interno, supponiamo di inserire i dati richiesti in 3 textbox e di effettuare la memorizzazione con un pulsante, il nostro codice sarà:

```

PUBLIC SUB Button2_Click()

    DIM sName AS String ' sName è una variabile che indica il nome che diamo al nostro database
    DIM sPercorso AS String ' indica il percorso in cui andremo a creare il db
    DIM $hConn AS NEW Connection
    DIM hres AS Result
    DIM sql AS String

    'controllo dati inseriti
    'sData = TextBox4.Text

    IF TextBox1.Text = "" OR TextBox2.text = "" OR TextBox3.text = "" OR TextBox4.text = "" THEN
    'controllo che siano inseriti dei valori nei campi richiesti
        Message.Error("Completare tutti i campi", "OK")

    ELSE
        sName = "prova"
        sPercorso = User.Home 'definiamo come percorso la nostra home

        WITH $hConn
            .Type = "sqlite3" ' indicazione del tipo di database da utilizzare

```

```
.Host = sPercorso
.Login = ""
.Password = ""
END WITH

$hConn.Name = sName
$hConn.Open

$hconn.Begin

    hres = $hconn.Create("biblio") 'collega il result hres alla tavola prova in modalità
creazione record

    hres!nome = TextBox1.Text 'scrittura dei vari campi del record
    hres!cognome = TextBox2.Text
    hres!indirizzo = TextBox3.Text

    hres.Update

    $hConn.Commit
    $hConn.Close

    textbox1.text=""
    textbox2.text=""
    textbox3.text=""

    label1.text = "aggiunta record effettuata"
ENDIF

END
```

## Ricerca dati

la ricerca dati sarà probabilmente l' operazione che verrà compiuta maggiormente in un db in quanto necessaria in altre operazioni come la modifica di un record o la cancellazione , in questo esempio effettueremo una ricerca per nome e visualizzeremo i dati in una GridView.

Aggiungiamo quindi al nostro form un textbox4 dove scriveremo il testo da ricercare ed una gridview dovrà quindi avere un aspetto come questo:



The screenshot shows a Windows Forms application window. On the left side, there is a vertical stack of controls: a button labeled 'Crea db', a label 'Label1', another button labeled 'Memorizza', and three text boxes labeled 'Nome', 'Cognome', and 'indirizzo'. On the right side, there is a section titled 'Ricerca per nome' containing a text box for search input and a GridView1 table with three rows and two columns. The table is currently empty.

Nella funzione **TextBox4\_Change()** inseriremo il codice per effettuare la ricerca, in questo modo scrivendo la parola da ricercare verranno visualizzate tutte le corrispondenze.

Logicamente prima di fare una ricerca dovremo aver inserito dei dati all' interno del db!

Come prima cosa dovremo collegarci al db tramite il solito metodo , una volta "connessi" per effettuare una ricerca utilizzeremo il metodo `connection.Exec` fornendogli una query SQL per la ricerca.

A seconda di ciò che vogliamo cercare useremo query diverse , nel nostro caso dobbiamo utilizzare una query che ricerchi nella colonna nome tutti i record che contengono il testo contenuto in `textbox4`:

```
sql = "SELECT * FROM biblio WHERE nome LIKE '%" & textbox4.Text & "%"
```

I caratteri % indicano un qualsiasi valore (carattere jolly)

quindi daremo la query in pasto al db:

```
hres = $hconn.Exec(sql)
```

Nel result "hres" sono presenti tutti i record che soddisfano la nostra ricerca, ora dobbiamo visualizzarli nella gridview , possiamo farlo con un ciclo FOR EACH su hres:

```
FOR EACH hres

    GridView1[i, 0].Text = hres!id
    GridView1[i, 1].Text = hres!nome
    GridView1[i, 2].Text = hres!cognome
    GridView1[i, 3].Text = hres!indirizzo

    i = i + 1

NEXT
```

Il primo valore nelle parentesi quadre indica il numero del record su cui scrivere ed il secondo il numero di colonna, i dati da estrapolare si indicano scrivendo il nome del result seguito da un punto esclamativo seguito dal nome del campo del db.

Oppure si può utilizzare questo tipo di codice :

```
FOR EACH hres

    GridView1[i, 0].Text = hres[0]
    GridView1[i, 1].Text = hres[1]
    GridView1[i, 2].Text = hres[2]
    GridView1[i, 3].Text = hres[3]

    i = i + 1

NEXT
```

Et voilà , mentre digitiamo una parola nel campo ricerca la gridview ci visualizzerà tutte le risposdenze, il codice completo della funzione sarà il seguente:

```
PUBLIC SUB TextBox4_Change()

    DIM sName AS String ' sName è una variabile che indica il nome che diamo al nostro database
    DIM sPercorso AS String ' indica il percorso in cui andremo a creare il db
    DIM $hConn AS NEW Connection
    DIM hres AS Result
    DIM sql AS String
    DIM i AS Integer
```

GridView1.Clear

```
sName = "prova"  
sPercorso = User.Home 'definiamo come percorso la nostra home
```

```
WITH $hConn  
.Type = "sqlite3" ' indicazione del tipo di database da utilizzare  
.Host = sPercorso  
.Login = ""  
.Password = ""  
END WITH
```

```
$hConn.Name = sName  
$hConn.Open
```

```
sql = "SELECT * FROM biblio WHERE nome LIKE '%" & textbox4.Text & "%"
```

' sql di ricerca segue questa sintassi :

```
'SELECT * FROM nome_tavola WHERE nome_colonna LIKE '%valore_da_cercare%'
```

```
hres = $hconn.Exec(sql) ' esegue la query
```

```
IF hres.Available = TRUE THEN ' controlla se la query ha dato dei risultati in caso  
' affermativo procede alla visualizzazione  
' altrimenti visualizza messaggio di errore
```

```
GridView1.Mode = Select.Multiple  
GridView1.Columns.Count = 4 'imposta il numero di colonne della GridView1  
GridView1.Rows.Count = hres.Count 'imposta il numero di record in base al numero  
' di risultati presenti
```

```
i = 0
```

```
FOR EACH hres 'inizio del ciclo
```

```
GridView1[i, 0].Text = hres!id 'scrive il 1° valore nella griglia  
GridView1[i, 1].Text = hres!nome  
GridView1[i, 2].Text = hres!cognome  
GridView1[i, 3].Text = hres!indirizzo
```

```
INC i 'aumenta di uno la variabile i in tal modo al  
'prossimo ciclo si scriverà il record successivo
```

```
NEXT
```

```
ELSE ' nel caso hres non contenga risultati
```

```
GridView1.Columns.Count = 1  
GridView1.Rows.Count = 1  
GridView1[0, 0].Text = "Nessun dato" 'viene visualizzata la scritta "nessun dato"
```

```
ENDIF
```

```
END
```



```
iRow = GridView1.Row
```

```
sql = "SELECT * FROM biblio WHERE nome LIKE '%" & textbox4.Text & "%"  
hres = $hconn.Exec(sql) ' esegue la query
```

```
IF iRow <> -1 THEN
```

```
    hres.MoveTo(iRow)  
    id = hres!id
```

```
    hres = $hConn.edit("biblio", "id=" & id)
```

```
    TextBox1.Text = hres!nome  
    TextBox2.Text = hres!cognome  
    TextBox3.Text = hres!indirizzo  
    Label6.Text = id
```

```
    Button2.Enabled = FALSE  
    Button3.Enabled = TRUE  
    Button4.Enabled = TRUE
```

```
ENDIF
```

```
$hConn.Close
```

```
END
```

```
PUBLIC SUB Button3_Click()
```

```
    DIM id AS String  
    DIM sName AS String ' sName è una variabile che indica il nome che diamo al nostro database  
    DIM sPercorso AS String ' indica il percorso in cui andremo a creare il db  
    DIM $hConn AS NEW Connection  
    DIM hres AS Result  
    DIM sql AS String
```

```
    sName = "prova"  
    sPercorso = User.Home 'definiamo come percorso la nostra home
```

```
    WITH $hConn  
        .Type = "sqlite3" ' indicazione del tipo di database da utilizzare  
        .Host = sPercorso  
        .Login = ""  
        .Password = ""  
    END WITH
```

```
    $hConn.Name = sName  
    $hConn.Open
```

```

id = Label6.Text

hres = $hConn.edit("biblio", "id=" & id)

hres!nome = TextBox1.Text
hres!cognome = TextBox2.Text
hres!indirizzo = TextBox3.Text

hres.Update

Button3.Enabled = FALSE
Button4.Enabled = FALSE
Button2.Enabled = TRUE

$hconn.Commit

TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
textbox4.Text = ""

$hConn.Close

```

END

### Cancellazione di un record

La cancellazione si effettua in modo simile alla modifica , si utilizza lo stesso metodo Connection.edit

```

PUBLIC SUB Button4_Click()

    DIM id AS String
    DIM sName AS String ' sName è una variabile che indica il nome che diamo al nostro database
    DIM sPercorso AS String ' indica il percorso in cui andremo a creare il db
    DIM $hConn AS NEW Connection
    DIM hres AS Result
    DIM sql AS String

    sName = "prova"
    sPercorso = User.Home 'definiamo come percorso la nostra home

    WITH $hConn
        .Type = "sqlite3" ' indicazione del tipo di database da utilizzare
        .Host = sPercorso
        .Login = ""

```

```
' .Password = ""  
END WITH  
  
$hConn.Name = sName  
$hConn.Open
```

```
id = Label6.Text
```

```
hres = $hConn.edit("biblio", "id=" & id)
```

```
hres.Delete
```

```
Button3.Enabled = FALSE  
Button4.Enabled = FALSE  
Button2.Enabled = TRUE
```

```
$hconn.Commit
```

```
TextBox1.Text = ""  
TextBox2.Text = ""  
TextBox3.Text = ""  
textbox4.Text = ""
```

```
$hConn.Close
```

```
END
```

THE END