

Introduzione alle pagine dinamiche

Con il linguaggio **HTML** si possono creare delle pagine **Web statiche**. Se invece volessimo creare delle pagine **Web dinamiche**, pagine il cui codice html viene generato al volo da un server, allora dovremo utilizzare il linguaggio **ASP** per la piattaforma NT-IIS e **PHP** per la piattaforma Linux-Apache.

Le tipiche applicazioni che utilizzano le pagine Web dinamiche, sono le interrogazioni a data base remoti.

Accesso ai DataBase da pagine ASP

Per effettuare la comunicazione da una pagina ASP al database si utilizza il componente ADO che consente l'accesso a tutti i tipi di dati. ADO mette a disposizione diversi oggetti che possono essere utilizzati per la connessione al database e per la sua interrogazione: l'oggetto **Connection**, **RecordSet**, **Error**, **Field**, **Command**.

- 1) l'oggetto **Connection** consente di stabilire la connessione con la sorgente di dati.
- 2) l'oggetto **RecordSet** consente di lavorare con i dati di una tabella, contenendo infatti un insieme di record della tabella stessa. Mediante questo oggetto possiamo leggere, modificare o aggiornare dati alla tabella.
- 3) l'oggetto **Command** combina l'oggetto RecordSet e l'oggetto Connectin.

Per estrarre dati da un database sono necessarie due fasi:

- Stabilire la connessione al database;
- Effettuare la vera e propria interrogazione al database.

Le pagine ASP.NET

Il codice ASP.NET è l'evoluzione del codice ASP classico ed è identificato dall'estensione **.aspx**.

È possibile eseguire pagine ASP e ASP.NET sullo stesso server Web senza interferenze. ASP.NET utilizza il linguaggio Visual Basic e la prima volta che il server riceve la richiesta di esecuzione di una pagina ASP.NET, questa viene compilata.

Per utilizzare le pagine ASP.NET il computer deve avere installato il S.O. Windows 2000 o XP o 2003 e un Web Server oltre a **.NET Framework** (librerie software di Microsoft per le applicazioni Web).

Accesso ai DataBase da pagine ASP.NET

ADO.NET è la tecnologia di accesso ai dati che rappresenta un'evoluzione di ADO; tale tecnica è utilizzabile con database di piccole e grandi dimensioni.

In **ADO.NET** non è più disponibile l'oggetto RecordSet.

Gli oggetti principali di **ADO.NET** per costruire applicazioni Web, con accesso ai database in rete, sono:

- 1) **Connection**, per stabilire la connessione al database
- 2) **Command**, per eseguire i comandi di manipolazione o interrogazione al database
- 3) **DataReader**, per ottenere i dati richiesti dal database.

I **DataReader** vengono poi associati ai **controlli server** di ADO.NET per visualizzare i dati in forma tabellare nelle pagine Web.

Tali controlli server sono:

- **DataGrid**
- **Repeater**
- **DataList.**

Il metodo **ExecuteReader** applicato all'oggetto di tipo *Command* esegue il comando SQL, memorizzato in una stringa, e crea i dati all'interno dell'oggetto *DataReader*. Esso viene associato poi, attraverso il metodo **DataBind**, al controllo definito nella pagina ASP.NET per la visualizzazione dei dati.

Per leggere e scrivere i dati occorre prima di tutto stabilire una connessione con il database attraverso un oggetto di tipo *Connection* che è diverso a seconda del database utilizzato: questo oggetto si chiama **SqlConnection** per i database *SQLServer* e **OleDbConnection** per i database *Access*. Lo spazio dei nomi (**namespace**) per *SqlServer* è `System.Data.SqlClient`, per *Access* è `System.Data.OleDb`.

Per questo motivo le pagine ASP.NET che accedono ai database di *Access* contengono come prima riga la dichiarazione di importazione degli oggetti *OleDb*:

```
<%@ import Namespace="System.Data.OleDb" %>
```

Struttura di una pagina ASP.NET

```
<%@ Page Language="VB" %>  
<script runat="server">  
  
    sub  
  
    end sub  
  
</script>  
  
<html>  
<head>  
</head>  
<body>  
    <form runat="server">  
  
        </form>  
</body>  
</html>
```

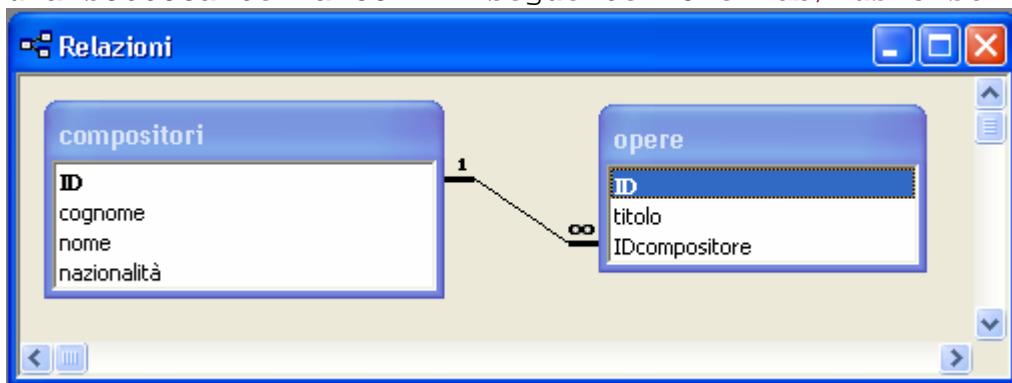
- 1) la prima parte, compresa tra `<script>` e `</script>`, raggruppa le istruzioni per l'elaborazione sul server. Il codice è organizzato in sottoprogrammi (`sub` `end sub`).
- 2) La seconda parte contiene gli elementi utilizzabili dal browser per presentare la pagina all'utente: tutti gli elementi sono raggruppati tra `<form>` e `</form>`.

Come gestire un data base on line con le pagine ASP.NET ESEMPIO PRATICO

Dopo aver creato il database (con Access) e creato il progetto utilizzando Visual Basic, si può gestire il database anche **on line**.

In una cartella del sito che si sta amministrando si può creare un file **index.html** col ruolo di gestore delle varie opzioni da scegliere. Per rendere tale file più semplice possibile, potremo immaginarlo con solo dei pulsanti che attivano le relative **pagine ASP.NET** idonee a gestire le specifiche richieste.

Ad esempio supponiamo di avere il seguente database memorizzato in una sottocartella con il seguente nome: **db/musicisti.mdb**.



e di voler realizzare:

- 1) una query parametrica che trova tutti i compositori di una nazione
- 2) ed una che è in grado di **inserire un nuovo compositore**.

La codifica in SQL è la seguente:

- 1)

```
SELECT compositori.cognome, compositori.nome,
compositori.nazionalità FROM compositori WHERE
(((compositori.cognome)=[inserisci la nazionalità]));
```
- 2)

```
Insert Into compositori (cognome, nome, nazionalità) Values
([inserire cognome],[inserire nome],[inserire nazionalità]);
```

La tecnologia ASP.NET e i database

Ecco quindi i relativi file per gestire ON LINE il BATABASE:

file index.html

```
<html>
<body>

  <form name="Pul_DataGrid" method="post"
  action="RicercaNazione_DataGrid.aspx">

    <input type="submit" name="Pul_visualizzaDG" value="Ricerca
    per Nazione_DataGrid">

  </form>

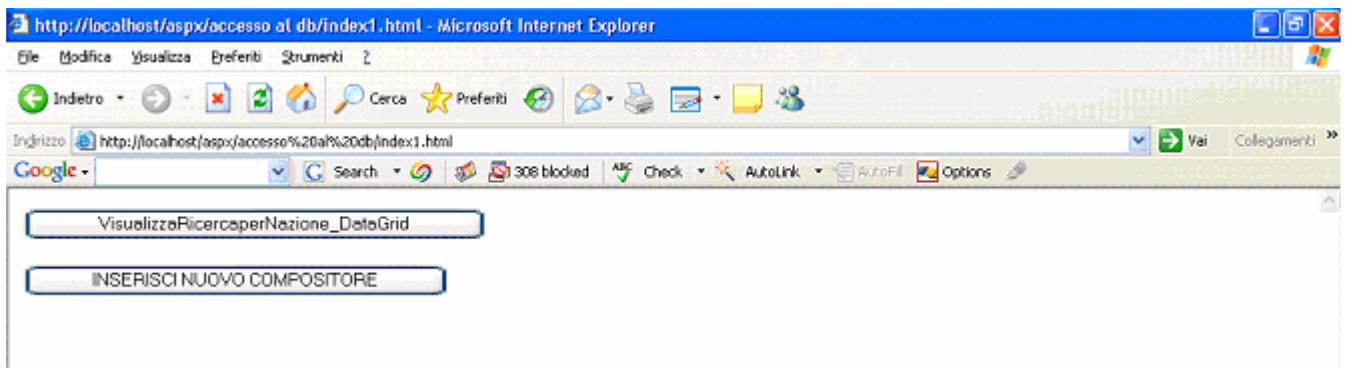
  <form name="Pul_Inserisci" method="post" action="inserisci.aspx">

    <input type="submit" name="Pulsante_inserisci" value="INSERISCI
    NUOVO COMPOSITORE">

  </form>

</body>
</html>
```

Ecco esecuzione di tale file:



File: RicercaNazione_DataGrid.aspx
Si utilizza il controllo DataGrid

Per realizzare la seguente query parametrica si utilizza il comando **SELECT** del linguaggio SQL per ottenere i dati da visualizzare. Ogni volta che si deve utilizzare il comando SELECT si può utilizzare tale pagina ASP.NET modificando solo la stringa strSQL.

```
<%@ Page Language="VB" %>
<%@ import Namespace="System.Data.OleDb" %>
<script runat="server">

    ' Insert page code here
    sub Page_Load
        lbl1.text="Ricerca Musicisti per Nazione "
    end sub

    Sub submit_click(sender As Object, e As EventArgs)
        Dim dbconn as OleDbConnection
        Dim dbcomm as OleDbCommand
        Dim dbread as OleDbDataReader
        Dim strSQL as string
        dim nome as string

        nome = NomeNazione.Text

        dbconn = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;
data source=" & server.mappath("db/musicisti.mdb"))
        dbconn.Open()

        strSQL = "Select cognome, nome From compositori "
        strSQL = strSQL & "where nazionalità = '" & nome & "' order by
cognome"

        dbcomm = New OleDbCommand(strSQL , dbconn)

        dbread = dbcomm.ExecuteReader()

        musici.DataSource=dbread
        musici.DataBind()
        dbread.Close()
        dbconn.Close()

    End Sub
</script>
```

```
<html>
<head>
</head>
<body>
  <form runat="server">
    <p>
      <asp:Label id="lbl1" runat="server">Ricerca</asp:Label>
    </p>
    <p>
      Quale Nazione:<asp:TextBox id="NomeNazione"
runat="server"></asp:TextBox>
    </p>
    <p>
      <asp:Button id="submit" onclick="submit_Click"
runat="server" Text="Cerca"></asp:Button>
    </p>
    <p>
    </p>
    <p>
      <asp:DataGrid id="musici" runat="server"

      headerstyle-font-name="Verdana"
      headerstyle-font-size="10pt"
      headerstyle-horizontalalign="center"
      headerstyle-font-bold="True"
      headerstyle-backcolor="#778899"
      headerstyle-forecolor="#ffffff"

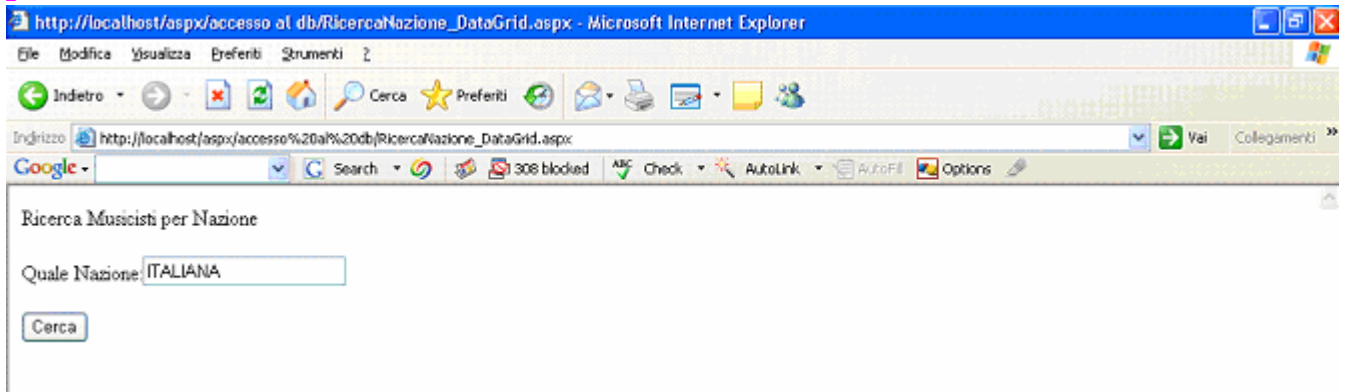
      >
      </asp:DataGrid>

    </p>
    <!-- Insert content here -->
  </form>
</body>
</html>
```

La tecnologia ASP.NET e i database

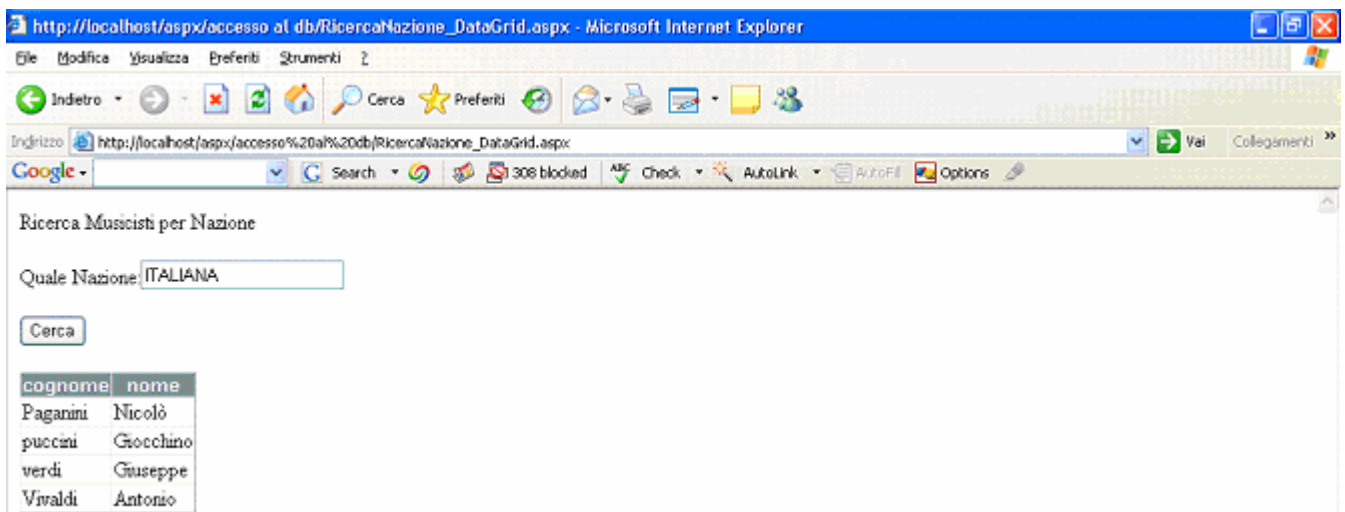
Ecco esecuzione di tale file:

prima videata:



seconda videata:

dopo aver cliccato il pulsante "Cerca"



File: **inserisci.aspx**

Con la seguente pagina ASP.NET possiamo usare i comandi di manipolazione **Insert**, **Update** e **Delete** del linguaggio SQL. L'esempio seguente mostra l'uso del comando Insert, ma la procedura presentata ha carattere generale in quanto, per gli altri comandi, basta sostituire la sola stringa contenente il comando SQL. L'esecuzione del comando SQL è attivata attraverso il metodo **ExecuteNonQuery**. Tale metodo inoltre restituisce un valore numerico che rappresenta il numero di righe (della tabella) interessate dal comandi SQL. Si deve osservare che, per consentire l'esecuzione dei comandi di manipolazione sui database in rete, l'Amministratore del Web server deve impostare i permessi di scrittura, sul file contenente il database, per gli utenti che accedono tramite pagine Web.

```
<%@ Page Language="VB" %>
<%@ import Namespace="System.Data.OleDb" %>
<script runat="server">

    sub Page_Load
        lbl1.text="Inserimento nuovo compositore "
    end sub

    Sub submit_click(sender As Object, e As EventArgs)

        Dim dbconn as OleDbConnection
        Dim dbcomm as OleDbCommand

        Dim strSQL as string
        dim numrec as integer

        dbconn = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;
data source=" & server.mappath("db/musicisti.mdb"))
        dbconn.Open()

        strSQL = "Insert Into compositori (cognome, nome, nazionalità) "
        strSQL = strSQL & "Values ("
        strSQL = strSQL & "'" & cognome.text & "', "
        strSQL = strSQL & "'" & nome.text & "', "
        strSQL = strSQL & "'" & nazionalità.text & "'"
        strSQL = strSQL & ")"

        dbcomm = New OleDbCommand(strSQL , dbconn)

        numrec= dbcomm.ExecuteNonQuery()
        lbl2.text = "Record aggiunti = " & numrec
        dbconn.Close()

    End Sub
```


La tecnologia ASP.NET e i database

