ACCESSO A UN DB

Dopo aver visto la sintassi del linguaggio PHP e le funzionalità di MySQL (per le interrogazioni in SQL di un DB) possiamo finalmente sfruttare questi due ambienti facendoli interagire fra loro al fine di permettere la costruzione di pagine Web dinamiche attraverso l'accesso a database in rete.
Ricordiamo che PHP è un linguaggio di scripting che estende le funzionalità di un Web Server. Mentre MySQL è un programma lato Server che si occupa di gestire i database di rete.
In un sito Web dinamico il contenuto delle pagine risiede in un database e ogni volta che un utente ne fa richiesta (attraverso il proprio sistema Client), questo contenuto viene recuperato e mostrato tramite le pagine Web realizzate con il linguaggio HTML.
Dal lato browser le informazioni devono essere presentate tramite marcatori (tag) del linguaggio HTML, mentre dal lato Server le stesse informazioni sono contenute in un database gestito da MySQL.
Le due parti si collegano attraverso script scritti in linguaggio PHP. I compiti svolti da questi script sono:

- Connessione al DB MySQL, invio dei comandi SQL e acquisizione delle risposte.
- Scrittura dei dati ricevuti dal DB all'interno di pagine Web in formato HTML, in modo che siano interpretate dal browser dell'utente.

La famiglia di funzioni mysql_*

In questa analisi si farà ricorso alle funzioni della famiglia mysql_*, ottime per uno studio prettamente didattico, ma considerate deprecate nelle versioni più recenti di PHP a favore delle più prestanti funzioni mysqli_*. Il passaggio a questa nuova famiglia di funzioni, fortunatamente, il è piuttosto semplice in quanto (mantenendo lo stile di programmazione procedurale) è sufficiente cambiare il prefisso "mysql_" in "mysqli_" per continuare ad utilizzare buona parte dei vecchi script.

@nomeFunzione()

Si ricordi che il simbolo "@" (silence) posto davanti al nome di una funzione impedisce alla funzione stessa di ritornare errori a schermo qualora non vada a buon fine. Si consiglia, quindi, di non utilizzarlo in fase di sviluppo, mentre è possibile metterlo in fase di produzione quando si è sicuri che il codice realizzato è corretto.

Array numerici e associati

Un array può essere di tre categorie in base alla tipologia di indice (chiave):

- Array numerico. La chiave per accedere ad un suo elemento è del tipo numerico.
- Array associativo. La chiave per accedere ad un elemento è del tipo stringa.
- Array numerico e associativo. La chiave può essere sia numerica che stringa.

APPROFONDIRE CON ESEMPI: http://www.html.it/pag/16688/gli-array2/

Il DB Rubrica

Il database che si utilizzerà per fare degli esempi sarà sempre il BD Rubrica creato seguendo i passaggi mostrati nella guida relativa a easyPHP.

Si analizzerà il modo con cui interrogare tale DB attraverso l'utilizzo di un form HTML e del linguaggio di scripting lato server PHP. In particolare, si realizzerà una pagina HTML contenente un form, attraverso il quale l'utente potrà inserire il cognome di una persona e premendo un bottone verrà inviata la richiesta di ricerca del relativo numero telefonico (possono esistere più persone con lo stesso cognome) ad uno script PHP. Lo script PHP elaborerà la richiesta accedendo al database nel quale sono contenuti i dati delle persone.

Si suppone di disporre, quindi, del solito **DB rubrica** di esempio costituito dalla seguente tabella **utenti**:

#	Nome	Tipo	Codifica caratteri	Attributi	Null	Predefinito	Extra
1	IdUtente	int(6)			No	*Nessuno*	AUTO_INCREMENT
2	Nome	varchar(30)	latin1_general_ci		No	*Nessuno*	
3	Cognome	varchar(30)	latin1_general_ci		No	*Nessuno*	
4	Tel	varchar(20)	latin1_general_ci		No	*Nessuno*	
5	Ruolo	varchar(20)	utf8_general_ci		No	*Nessuno*	

Lo schema relazionale è il seguente:

`utenti(IdUtente, Nome, Cognome, Tel, Ruolo)`

dove IdUtente è la chiave primaria autoincrementale numerica.

La tabella contiene i seguenti record di esempio:

IdUtente chiave primaria autoincrementale	Nome	Cognome	Tel	Ruolo
1	daniele	verdi	0331101011	amico
2	paolo	rossi	0222334433	parente
3	riccardo	rossi	0332404040	amico
4	riccardo	bianchi	0660606044	amico
5	luca	verdi	0612345655	parente
6	emma	verdi	0331776655	amico
7	lucia	gialli	0331998877	conoscente
8	mario	rossini	0677665511	amico
9	luca	rossini	0677665511	conoscente
10	luca	colombo	1230987	parente
11	anna	grassi	1230987	parente
13	lucio	dalla	545355	amico
14	paolo	neri	1204567	conoscente

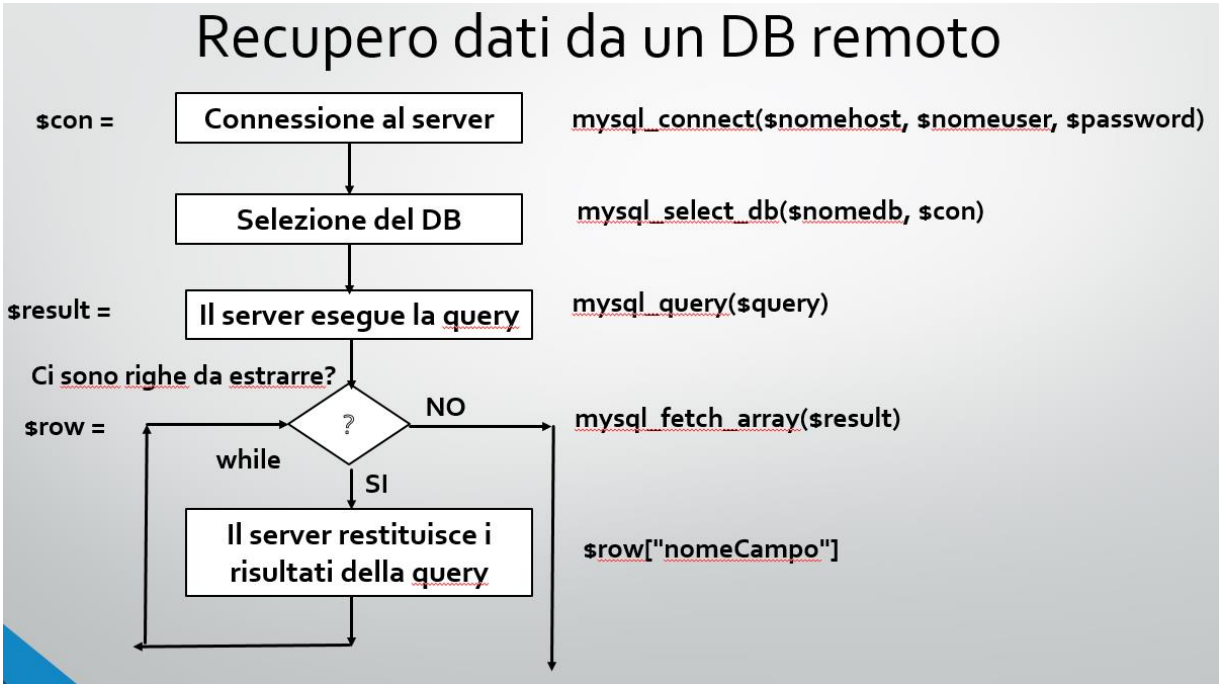
INVIO DATI AL SERVER TRAMITE UN FORM HTML

Uno degli aspetti più importanti di un linguaggio di scripting è quello di poter far interagire l'utente con l'applicazione Web. L'interazione in PHP è realizzata, per esempio, facendo inserire all'utente i dati in un form HTML, inviarli al Web Server e far si che lo script lato server riesca a elaborare la richiesta generando una risposta HTML in real time da far visualizzare al browser dell'utente stesso. In questo modo si realizza una pagina Web dinamica.

L'obiettivo del seguente esempio è quello di cercare il numero di telefono di un utente registrato nel DB rubrica.

Il progetto da realizzare si compone di due file:

- La pagina (formCercaTel.htm) contenente il form per acquisire i dati dell'utente.
- La pagina PHP (elaboraCercaTel.php) richiamata dalla pagina HTML, che riceve i dati e ricerca l'utente nel DB che ha quel numero di telefono.



LATO CLIENT

Il lato client dell'applicazione è quindi costruita semplicemente da un form (modulo) HTML costituito da una casella di testo (text) e da un pulsante (submit). La pagina HTML da realizzare è quindi la seguente:

formCercaTel.htm

```
<!DOCTYPE html>
<html>
<head>
      <title>RUBRICA 2015</title>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
    <form action="elaboraCercaTel.php" method="post">
            Cerca Telefono per Cognome
            <input type="text" name="cognome" size="30" maxlength="30" />
            <br />
            <input type="submit" value="CERCA" />
      </form>
</body>
</html>
```

Si visualizzi la pagina digitando nella barra degli indirizzi di Firefox il seguente URL:

http://localhost/DBrubrica/cercaTel.htm

La pagina HTML è costituita da un form realizzato attraverso il tag `<form> elementi </form>`. Fra il tag di apertura e di chiusura si includono tutti gli elementi che si desiderano visualizzare.

Il tag form ha i seguenti attributi:

- `action="elaboraCercaTel.php"`: specifica la pagina PHP che si incaricherà di elaborare la richiesta.
- `method="post"`: specifica la modalità con la quale i dati verranno inviati alla pagina PHP.

Gli elementi contenuti nel form sono:

- Un campo d'input, rappresentato con il tag `<input type="text">`, utilizzato per permettere all'utente di inserire una stringa di caratteri (per esempio, nel nostro caso, il cognome di una persona).
- Un bottone, rappresentato con il tag `<input type="submit">;` cliccando su questo bottone i dati verranno inviati alla pagina PHP.

Il tag `<input>` possiede i seguenti attributi:

- `type="text"`: identifica il tipo di campo, di testo, appunto.
- `name="cognome"`: identifica il nome del campo; questo nome verrà utilizzato come variabile a cui assegnare un valore (il cognome di una persona)da passare mediante il metodo post alla pagina di elaborazione PHP.
- `size="30"`: dimensione del campo.
- `maxlength="30"`: numero massimo di caratteri digitabili nel campo.

il tag button possiede i seguenti attributi:

- `type="submit"`: specifica che si tratta di un bottone.
- `value="CERCA"`: è l'etichetta visualizzata sul bottone.

Premendo, quindi, il tasto `CERCA`, il browser richiama lo script `elaboraCercaTel.php` indicato nell'intestazione del form come valore dell'attributo **action.** Il browser aggiunge automaticamente alla richiesta tutti i campi presenti nel form. Per ogni campo crea un parametro avente come nome il nome del campo (indicato dall'attributo **name** del tag <**input**>) e come valore ciò che l'utente inserisce nella casella di testo. Nel nostro esempio viene creato il parametro `cognome` con il relativo valore, e questi vengono inviati allo script attraverso il metodo post (utilizzato in genere per spedire grandi quantità di dati non visibili a nessun utente della rete).

LATO SERVER

Il lato server dell'applicazione Web è costituita dalla seguente pagina PHP:

VER 1.0

elaboraCercaTel.php

Ver 1.0 - Visualizzazione dati recuperati	
Riga	**Codice**
1	`<?php`
2	`    // parametri per la connessione al database`
3	`    $nomehost = "localhost"; //hostname o indirizzo IP del server`
4	`    $nomeuser = "root"; //nome utente per la connessione a MySQL`
5	`    $password = "admin"; //password per la connessione a MySQL`
6	`    $nomedb = "rubrica"; //nome del database MySQL`
7	`    //connessione al Server tramite mysql_connect`
8	`    $con = mysql_connect($nomehost, $nomeuser, $password)`
9	`    or die ("Impossibile connettersi al Server ...");`
10	`    //selezione del DB`
11	`    mysql_select_db($nomedb, $con)`
12	`    or die ("Impossibile connettersi al DB ...");`
13	`    echo(" Access to DB" . $nomedb);`
14	`    /*recupero il valore del parametro di ricerca cognome passato`
15	`      con metodo POST */`

```
16      $cognome = $_POST["cognome"];
17      //creo e salvo la query nella variabile stringa $query
18      $query = "SELECT * FROM utenti WHERE Cognome = '$cognome'";
19      $result = mysql_query($query) or die("Error Query...");
20      //eseguo la query
21      //numero di righe restituite dalla query
22      $num_row = mysql_num_rows($result);
23      //numero di campi restituiti dalla query
24      $num_col = mysql_num_fields($result);
25      print("<br />" . "Numero di righe trovate: " . $num_row);
26      echo("<br />" . "Numero di campi trovati: " . $num_col);
27      printf("<h2>Dati trovati</h2>");
28      /*carico in row una riga del vettore associativo usando i nomi
29        dei campi restituiti dalla query*/
30      while($row = mysql_fetch_array($result)){
31          echo("<p>");
32          echo("Cognome: " . $row["Cognome"]);
33          echo("   -- ");
34          echo("Nome: ".$row["Nome"]);
35          echo("   -- ");
36          echo("Tel: ".$row["Tel"]);
37          echo("</p>");
38      }
39      mysql_close($con); //chiudo la connessione
40  ?>
```

In un sito Web dinamico il contenuto delle pagine risiede in un database e ogni volta che un utente ne fa richiesta, questo contenuto viene recuperato e mostrato tramite le pagine Web realizzate con il linguaggio HTML.

Dal lato client le informazioni devono essere presentate tramite marcatori (tag) del linguaggio HTML, mentre dal lato server le stesse informazioni sono contenute in un database gestito da mySQL.

Abbiamo già analizzato come Client e Web Server interagiscono per comunicare e permettere all'utente di visualizzare sul proprio browser le informazioni richieste. Vediamo ora più in dettaglio quali sono i compiti svolti dalle due parti (client e server) quando si collegano usando gli script in linguaggio PHP:

1. Il client, attraverso il browser, effettua la richiesta di una pagina web dinamica al Web Server in formato PHP.
2. Il Web Server riconosce che si tratta di una richiesta PHP e quindi attiva l'interprete PHP per eseguire gli script in essa contenuti.
3. Se lo script contiene i comandi per collegarsi al database mySQL contenuto nel Server mySQL, il Web Server effettua una connessione a tale Server database.
4. Viene selezionato il database su cui si vogliono fare delle interrogazioni.
5. Il Web Server invia le richieste tramite comandi in SQL.
6. Il Server mySQL risponde restituendo al Web Server i dati richiesti.
7. Il Web Server risponde al Client generando dinamicamente una pagina HTML, grazie sempre allo script PHP, in cui sono presenti i dati richiesti.

Analizziamo quindi lo script precedente è costituito dai seguenti blocchi funzionali:

- APERTURA DELLA CONNESSIONE.

  Alla riga 8 viene invocata la funzione

  `mysql_connect($nomehost, $nomeuser, $password)`

  La funzione mysl_connect() apre una connessione con il Server mySQL.

  La funzione riceve tre parametri (argomenti/variabili d'ingresso):

  - `$nomehost`: rappresenta l'indirizzo IP o il nome del server su cui è in esecuzione il Server mySQL; se è lo stesso server su cui è in esecuzione il Web Server, si indica con `localhost`.
  - `$nomeuser`: è il nome dell'utente che ha i permessi per accedere al database.
  - `$password`: è la password dell'utente che ha i permessi per accedere al database.

  La funzione restituisce un valore, che viene salvato nella variabile `$con`, che corrisponde ad un numero ovveo all'**identificativo della connessione** nel caso in cui la connessione ha avuto successo. Se non è stata in grado di aprire una connessione, vedi riga 9

  ```
  or die("Impossibile connettersi al Server ...");
  ```

  viene restituito FALSE, che viene utilizzato per interrompere l'esecuzione e mostrare un messaggio di errore all'utente.

- SELEZIONE DEL DATABASE

  Alla riga 11 viene invocata la funzione

`mysql_select_db($nomedb, $con)`

per selezionare un database già esistente sul Server mySQL.

La funzione possiede due argomenti:

- `$nomedb`: è il nome del database che si vuole selezionare.
- `$con`; è l'identificativo restituito dalla funzione `mysql_connect()`.

La funzione restituisce un valore booleano. Se è TRUE indica che l'operazione è andata a buon fine ed è quindi possibile interagire con il database tramite comandi SQL. Se è FALSE indica che l'operazione non è andata a buon fine e quindi viene mostrato un messaggio di errore (riga 13).

- RECUPERO DATI DAL FORM

  Nella riga 16 viene recuperato il valore che l'utente ha scritto nel campo di input attraverso l'array associativo `$_POST`. Come indice di questo array occorre indicare il nome del campo di input in cui l'utente inserisce il cognome dell'utente/i da ricercare.

  ```
  $cognome = $_POST["cognome"];
  ```

  Salviamo nella variabile `$cognome` il valore recuperato dal form al fine di poterlo utilizzare nello script.

- GENERO LA STRINGA DI INTERROGAZIONE

  Nella riga 18 viene salvata nella variabile `$query` la stringa SQL per effettuare l'interrogazione che ci necessita: recupero tutti i record il cui campo Cognome coincide con il criterio di ricerca.

  ```
  $query = "SELECT * FROM utenti WHERE Cognome = '$cognome'";
  ```

  l'asterisco * indica che vengono selezionate tutte le colonne della tabella.

- INVIO DELLA STRINGA AL SERVER MYSQL

  Nella riga 19 viene invocata la funzione

  `mysql_query($query)`

  che consente di inviare la stringa d'interrogazione al Server mySQL.

  La funzione ha come unico argomento la stringa SQL e restituisce nella variabile `$result` un valore, che corrisponde all'**identificativo di risorsa** nel caso in cui l'invio ha avuto successo, utilizzato per indicare se l'invio è andato a buon fine altrimenti viene visualizzato un messaggio d'errore.

- RECUPERO IL NUMERO DI RIGHE E DI CAMPI RESTITUITI DALLA QUERY

  Nelle righe 22 e 24 le funzioni

```
mysql_num_rows($result);
```

```
mysql_num_fields($result);
```

restituiscono rispettivamente il numero di righe `$num_row` e il numero di campi estratti `$num_col` dalla query SQL. Alla funzione viene passato come argomento l'identificativo di risorsa.

Queste due variabili `$num_row` e `$num_col` risulteranno molto utili come indicatori del numero di operazioni ripetitive da eseguire (es lettura dei `$num_row` record).

- ESTRAZIONE DELLE RIGHE RESTITUITE DALLA QUERY

  Nella riga 30 viene invocata (tante volte quante sono le righe restituite) la funzione

  `mysql_fetch_array($result))`

  che riceve come argomento l'identificativo di risorsa e restituisce un array, `$row`, corrispondente alla riga successiva del risultato. In altri termini nel vettore `$row` verrà salvato la riga successiva estratta dalla query. Pertanto se questa funzione viene richiamata tante volte quante sono le righe restituite dalla query, allora è stato individuato un modo per estrarre e salvare tutti i record restituiti dalla query.

  Nella stessa riga 30, infatti, si può notare il ciclo WHILE che viene eseguito fin tanto che c'è una riga da poter leggere.

- VISUALIZZAZIONE DEI RISULTATI

  Il corpo del ciclo WHILE, tra la riga 31 e la riga 37, consente la visualizzazione dei dati in forma grezza (non tabellati come si vredà nella successiva versione dell'applicazione).

- CHIUSURA DELLA CONNESSIONE ATTIVA

  Nella riga 39 viene invocata la funzione che consente di chiudere la connessione attiva.

  `mysql_close($con);`

Questo è il risultato visivo:

Access to DBrubrica
Numero di righe che corrispondono al criterio di ricerca: 2
Numero di campi restituiti dalla query: 4

Dati trovati

Cognome = rossi - Nome = paolo - Tel = 02223344

Cognome = rossi - Nome = luca - Tel = 0332404040

2	paolo	rossi	02223344
3	luca	rossi	0332404040

VER 2.0

Miglioriamo il precedente script tabulando i dati e facendo uso del costrutto recordset che consente di salvare in una tabella virtuale tutti i dati restituiti dall'esecuzione sul server della query.

Ver 2.0 - Visualizzazione tabellare con recordset

Riga	Codice
1	`<?php`
2	`    // parametri per la connessione al database`
3	`    $nomehost = "localhost"; //hostname o indirizzo IP del server`
4	`    $nomeuser = "root"; //nome utente per la connessione a MySQL`
5	`    $password = "admin"; //password per la connessione a MySQL`
6	`    $nomedb = "rubrica"; //nome del database MySQL`
7	`    //connessione al Server tramite mysql_connect`
8	`    $con = mysql_connect($nomehost, $nomeuser, $password)`
9	`    or die("Impossibile connettersi al Server ...");`
10	`    //selezione del DB`
11	`    mysql_select_db($nomedb, $con)`
12	`    or die("Impossibile connettersi al DB ...");`
13	`    /*recupero il valore del parametro di ricerca cognome passato`
14	`      con metodo POST */`
15	`    $cognome = $_POST["cognome"];`
16	`    //creo e salvo la query nella variabile stringa $query`
17	`    $query = "SELECT * FROM utenti WHERE Cognome = '$cognome'";`
18	`    $result = mysql_query($query) or die ("Error Query...");`
19	`    //eseguo la query`

```
20      //numero di righe restituite dalla query
21      $num_row = mysql_num_rows($result);
22      //numero di campi restituiti dalla query
23      $num_col = mysql_num_fields($result);
24      printf("<h2>Dati trovati</h2>");
25      //metodo 2 - visualizzazione tabellare dei dati
26      //salvo le righe nel oggetto recordset
27      while($recordset[] = mysql_fetch_array($result));
28      echo("<table border=\"1\">");
29      echo("<tr>");
30      for($icol=0; $icol < $num_col; $icol++){
31          echo("<td>" . mysql_field_name($result, $icol) . "</td>");
32      }
33      echo("</tr>");
34      for($irow=0; $irow < $num_row; $irow++){
35          echo("<tr>");
36          for($icol=0; $icol < $num_col; $icol++){
37              echo("<td>" . $recordset[$irow][$icol] . "</td>");
38          }
39          echo("</tr>");
40      }
41      echo("</table>");
42      mysql_close($con); //chiudo la connessione
43  ?>
```

Le novità rispetto alla precedente versione sono:

- SALVATAGGIO DELLE RIGHE RESTITUITE DALLA QUERY NEL RECORDSET

  Nel blocco WHILE (riga 26) salviamo nel vettore `$recordset` (vettore a due dimensioni, ovvero una matrice) tutte le righe (record) restituiti dalla query. Questo vettore potrà quindi essere utilizzato per fare operazioni di visualizzazione, ricerca, modifica, etc.

- RESTITUZIONE NOME DEI CAMPI

  Nella riga 31 viene invocata la funzione

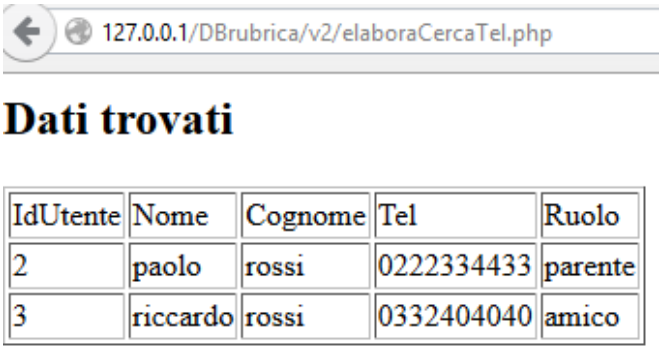
  `mysql_field_name($result, $icol)`

  che restituisce il nome del campo specificato dal risultato utilizzato per visualizzare le intestazioni di colonna. Alla funzione vengono passati due argomenti, il risultato (identificativo di risultato) e l'indice del campo.

- VISUALIZZAZIONE DATI IN FORMA TABELLARE

  Nel blocco FOR (dalla riga 34 alla riga 39) visualizzo in una tabella i risultati della query.

Il risultato visivo è il seguente:



VER 3.0

I blocchi relativi alla configurazione e alla connessione del DB spesso vengono ripetuti in ogni file dell'applicazione in cui si intenda connettersi al DB per reperire dati. È possibile, quindi, migliorare ulteriormente la precedente versione, scorporando dal file, questi blocchi che si ripetono facendo uso della funzione **include()**. Sarebbe uno spreco di tempo e di codice dover scrivere lo stesso codice in più file, con il rischio che se tale blocco contiene degli errori occorrerebbe correggerlo in tutti i file. La funzione **include()** e una duale **require()** servono proprio a questo scopo. Scrivere i blocchi di codice che si ripetono in un file separato e includerlo tutte le volte che lo si vuole utilizzare nel file corrente.

Inoltre è possibile verificare il caso anomalo in cui la ricerca non ha dato esiti positivi, cioè il caso in cui non esistono righe restituite dalla query.

Ver 3.0 - Visualizzazione classica tabellare con recordset e con l'individuazione di nessuno o più righe risultanti	
Riga	**Codice**
	elaboraCercaTel.php
1	`<?php`
2	`include("conn_db.php");`
3	`//includo il file di configurazione e connessione al DB`
4	`$cognome = $_POST["cognome"];`
5	`$query = "SELECT * FROM utenti WHERE Cognome = '$cognome'";`

```
6       $result = mysql_query($query) or die ("Error Query...");
7       $num_row = mysql_num_rows($result);
8       $num_col = mysql_num_fields($result);
9       if($num_row == 0){
10            echo 'nessun risultato di ricerca';
11      }
12      else{
13            while($recordset[] = mysql_fetch_array($result));
14            echo("<table border=\"1\">");
15            echo("<tr>");
16            for($icol=0; $icol < $num_col; $icol++){
17                  echo("<td>" .
18                  mysql_field_name($result, $icol) . "</td>");
19            }
20            echo("</tr>");
21            for($irow=0; $irow < $num_row; $irow++){
22                  echo("<tr>");
23                  for($icol=0; $icol < $num_col; $icol++){
24                        echo("<td>" .
25                        $recordset[$irow][$icol] . "</td>");
26                  }
27                  echo("</tr>");
28            }
29            echo("</table>");
30      }
31    mysql_close($con);
32  ?>
```

config.php

```
1   <?php
2       //par connessione al DB su server virtuale easyphp
3       $nomehost="localhost"; //hostname
4       $nomeuser="root"; //nome utente per la connessione a MySQL
5       $password="admin"; //password per la connessione a MySQL
6       $nomedb="rubrica"; //nome del DB MySQL
7   ?>
```

conn_db.php

```
1   <?php
```

```
2       //includo file configurazione per recupero variabili
3       include("config.php");
4       //connessione al Server tramite mysql_connect
5       $con=mysql_connect($nomehost,$nomeuser,$password) or
6       die("Impossibile connettersi al Server ...");
7       //connessione al DB
8       $db_selected=mysql_select_db($nomedb, $con) or
9       die("Errore nella selezione del database ...");
10  ?>
```

INCLUSIONE

Alla riga 2 del file elaboraCercaTel.php viene richiamata la funzione

```
include("conn_db.php");
```

per includere nel file corrente il codice in esso contenuto.

Alternativa molto simile è quella che utilizza la funzione require():

```
require("conn_db.php");
```

L'unica differenza fra le due funzioni consiste nella gestione di eventuali errori: nel caso il file da includere non si trovato `include()` genererà un warning mentre `require()` un fatal error (bloccando, di fatto, l'esecuzione dello script).

Si noti che, affinché l'inclusione vada a buon fine, è necessario specificare il percorso corretto del file che si desidera includere (nel nostro esempio il file "`conn_db.php`" si trova nella stessa cartella degli script che lo includono).

VER 4.0 – VERSIONE FINALE

Miglioriamo ulteriormente.

Ver 4.0 - Visualizzazione classica tabellare senza recordset	
Riga	**Codice**
	elaboraCercaTel.php
1	`<?php`
2	`include("conn_db.php");`
3	`//includo il file di configur. e connessione al DB sul server`
4	`$cognome = $_POST["cognome"];`

```
5       $query = "SELECT * FROM utenti WHERE Cognome = '$cognome'";
6       $result = mysql_query($query) or die ("Error Query...");
7       $num_row = mysql_num_rows($result);
8       $num_col = mysql_num_fields($result);
9       if($num_row == 0){
10            echo 'nessun risultato di ricerca';
11      }
12      else{
13            echo("<table border=\"1\">");
14            echo("<tr>");
15                  echo("<td><b>IdUtente</b></td>");
16                  echo("<td><b>Nome</b></td>");
17                  echo("<td><b>Cognome</b></td>");
18                  echo("<td><b>Tel</b></td>");
19                  echo("<td><b>Ruolo</b></td>");
20            echo("</tr>");
21            while($row = mysql_fetch_array($result)){
22                  echo("<tr>");
23                        echo("<td>" . $row["IdUtente"] . "</td>");
24                        echo("<td>" . $row["Nome"] . "</td>");
25                        echo("<td>" . $row["Cognome"] . "</td>");
26                        echo("<td>" . $row["Tel"] . "</td>");
27                        echo("<td>" . $row["Ruolo"] . "</td>");
28                  echo("</tr>");
29            }
30            echo("</table>");
31      }
32    mysql_close($con);
33  ?>
```

config.php

```
1   <?php
2       //par connessione al DB su server virtuale easyphp
3       $nomehost="localhost"; //hostname
4       $nomeuser="root"; //nome utente per la connessione a MySQL
5       $password="admin"; //password per la connessione a MySQL
6       $nomedb="rubrica"; //nome del DB MySQL
7   ?>
```

conn_db.php

```
1   <?php
2        //includo file configurazione per recupero variabili
3        include("config.php");
4        //connessione al Server tramite mysql_connect
5        $con=mysql_connect($nomehost,$nomeuser,$password) or
6        die("Impossibile connettersi al Server ...");
7        //connessione al DB
8        $db_selected=mysql_select_db($nomedb, $con) or
9        die("Errore nella selezione del database ...");
10  ?>
```

ALTERNATIVE A mysql_fetch_array()

`mysql_fetch_row()` - Recupera il contenuto dei records trovati. Più precisamente restituisce una array contenente i valori di ogni campo riscontrato nel recordset che si potrebbe poi richiamare specificando il relativo indice numerico.

```
// usando mysql_fetch_rows() si può scrivere ...
while($row = mysql_fetch_row($result)){
      $IdUtente=$row[0];
      $Nome=$row[1];
      $Cognome=$row[2];
      $Tel=$row[3];
      $Ruolo=$row[4];
      ……
}
```

`mysql_fetch_assoc()` - Recupera i dati dal recordset mediante l'indicazione del nome del campo (invece che dell'indice numerico).

```
// ...usando mysql_fetch_assoc() avremmo scritto
$Cognome=$row["Cognome"];
```

`mysql_fetch_array()` - Supporta, indistintamente, entrambe le tecniche di chiamata.

VER 5.0 – VER FINALE SEMPLIFICATA

Miglioriamo ulteriormente semplificando il codice.

Ver 5.0 - Visualizzazione classica tabellare nella versione finale

elaboraCercaTel.php

```
1  <?php include("conn_db.php");
2  //includo il file di configurazione e connessione al DB sul server
3      $cognome = $_POST["cognome"];
4      $query = "SELECT * FROM utenti WHERE Cognome = '$cognome'";
5      $result = mysql_query($query) or die ("Error Query...");
6      $num_row = mysql_num_rows($result);
7      $num_col = mysql_num_fields($result);
8      if($num_row == 0){
9        echo 'nessun risultato di ricerca';
10     }
11     else{
12       echo("<table border=\"1\">");
13       echo("<tr>");
14       for($icol=0; $icol < $num_col; $icol++)
15             echo("<td>" . mysql_field_name($result, $icol) . "</td>");
16       echo("</tr>");
17       while($row = mysql_fetch_array($result)){
18             echo("<tr>");
19             for($icol=0; $icol < $num_col; $icol++)
20                   echo("<td>" . $row[$icol] . "</td>");
21             echo("</tr>");
22       }
23       echo("</table>");
24     }
25     mysql_close($con);
26 ?>
```

config.php

```
1  <?php
2        //par connessione al DB su server virtuale easyphp
3        $nomehost="localhost"; //hostname
4        $nomeuser="root"; //nome utente per la connessione a MySQL
5        $password="admin"; //password per la connessione a MySQL
6        $nomedb="rubrica"; //nome del DB MySQL
7  ?>
```

conn_db.php

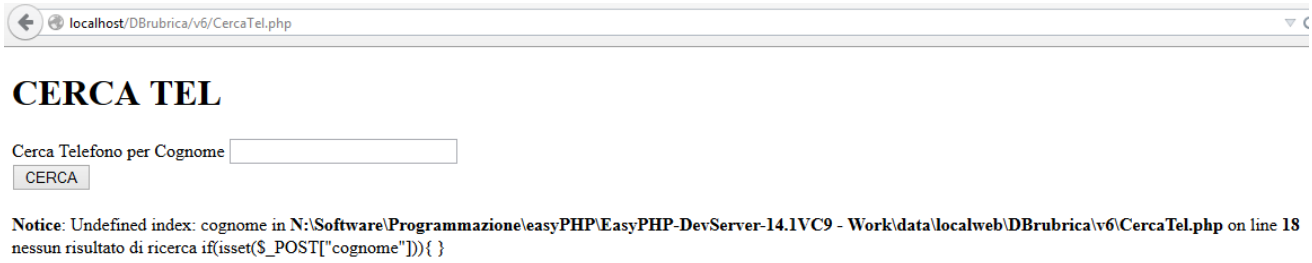
```
1   <?php
2        //includo file configurazione per recupero variabili
3        include("config.php");
4        //connessione al Server tramite mysql_connect
5        $con=mysql_connect($nomehost,$nomeuser,$password) or
6        die("Impossibile connettersi al Server ...");
7         //connessione al DB
8         $db_selected=mysql_select_db($nomedb, $con) or
9        die("Errore nella selezione del database ...");
10  ?>
```

VER 6.0 – TUTTO IN UN SOLO FILE PHP

Riga	Codice
	cercaTel.php

```
1   <!DOCTYPE html>
2   <html>
3   <head>
4         <title>RUBRICA 2015</title>
5         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
6   </head>
7   <body>
8       <h1>CERCA TEL</h1>
9         <form action="CercaTel.php" method="post">
10              Cerca Telefono per Cognome
11              <input type="text" name="cognome" size="30" maxlength="30" />
12              <br />
13              <input type="submit" value="CERCA" />
14        </form>
15
16  <?php include("conn_db.php");
17  //includo il file di configurazione e connessione al DB sul server
18      $cognome = $_POST["cognome"];
19      $query = "SELECT * FROM utenti WHERE Cognome = '$cognome'";
20      $result = mysql_query($query) or die ("Error Query...");
21      $num_row = mysql_num_rows($result);
22      $num_col = mysql_num_fields($result);
```

```
23      if($num_row == 0){
24        echo 'nessun risultato di ricerca';
25      }
26      else{
27        echo("<table border=\"1\">");
28        echo("<tr>");
29        for($icol=0; $icol < $num_col; $icol++)
30              echo("<td>" . mysql_field_name($result, $icol) . "</td>");
31        echo("</tr>");
32        while($row = mysql_fetch_array($result)){
33              echo("<tr>");
34              for($icol=0; $icol < $num_col; $icol++)
35                    echo("<td>" . $row[$icol] . "</td>");
36              echo("</tr>");
37        }
38        echo("</table>");
39      }
40      mysql_close($con);
41  ?>
42
43  </body>
44  </html>
```



Allora si utilizza ISSET()

```
<!DOCTYPE html>
<html>
<head>
      <title>RUBRICA 2015</title>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
    <h1>CERCA TEL</h1>
      <form action="CercaTel.php" method="post">
            Cerca Telefono per Cognome
```

```
            <input type="text" name="cognome" size="30" maxlength="30" />
            <br />
            <input type="submit" value="CERCA" />
      </form>

<?php
    if(isset($_POST["cognome"])){
            //includo il file di configurazione e connessione al DB sul server
            include("conn_db.php");
            $cognome = $_POST["cognome"];
            $query = "SELECT * FROM utenti WHERE Cognome = '$cognome'";
            $result = mysql_query($query) or die ("Error Query...");
            $num_row = mysql_num_rows($result);
            $num_col = mysql_num_fields($result);
            if($num_row == 0){
                  echo 'nessun risultato di ricerca';
            }
            else{
                  echo("<table border=\"1\">");
                  echo("<tr>");
                  for($icol=0; $icol < $num_col; $icol++)
                        echo("<td>" .
                              mysql_field_name($result, $icol) . "</td>");
                  echo("</tr>");
                  while($row = mysql_fetch_array($result)){
                        echo("<tr>");
                        for($icol=0; $icol < $num_col; $icol++)
                              echo("<td>" . $row[$icol] . "</td>");
                        echo("</tr>");
                  }
                  echo("</table>");
            }
            mysql_close($con);
      }
?>

</body>
</html>
```

VER 7.0 – TUTTO IN UN SOLO FILE PHP CON FILTRO

L'utente non deve inserire il cognome da ricercare, ma viene visualizzato in una select prelevando i cognomi dal DB.

```
<!DOCTYPE html>

<html>
<head>
	<title>RUBRICA 2015</title>
	<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
    <h1>CERCA TEL</h1>
	<form action="CercaTel.php" method="post">
		Cerca Telefono per Cognome
<?php
		include("conn_db.php");
		$query = "SELECT DISTINCT Cognome FROM utenti ORDER BY Cognome";
		$result = mysql_query($query) or die ("Error Query...");
		$num_row = mysql_num_rows($result);
		while($row = mysql_fetch_array($result))
			$rowCurr[] = $row[0];
		mysql_close($con);
		echo '<select name="cognome">';
		for($i=0; $i<$num_row; $i++){
			echo '<option value="' . $rowCurr[$i] . '">' . $rowCurr[$i] .
'</option>';
		}
		echo '</select>';
?>
		<br />
		<input type="submit" value="CERCA" />
	</form>

<?php
	if(isset($_POST["cognome"])){
		include("conn_db.php");
		$cognome = $_POST["cognome"];
		$query = "SELECT * FROM utenti WHERE Cognome = '$cognome' ORDER BY
Cognome, Nome";
```

```
            $result = mysql_query($query) or die ("Error Query...");
            $num_row = mysql_num_rows($result);
            $num_col = mysql_num_fields($result);
            echo("<table border=\"1\">");
            echo("<tr>");
            for($icol=0; $icol < $num_col; $icol++)
                  echo("<td>" . mysql_field_name($result, $icol) . "</td>");
            echo("</tr>");
            while($row = mysql_fetch_array($result)){
                  echo("<tr>");
                        for($icol=0; $icol < $num_col; $icol++)
                              echo("<td>" . $row[$icol] . "</td>");
                  echo("</tr>");
            }
            echo("</table>");
            mysql_close($con);
      }
?>

</body>
</html>
```

VER 8.0 – TUTTO IN UN SOLO FILE PHP CON FILTRO E FUNZIONI

```
<?php
      function estraiRecord(&$num_row, &$num_col, $query){
            //estrae nella prima riga anche l'intestazione
            include("conn_db.php");
            $result = mysql_query($query) or die ("Error Query...");
            $num_row = mysql_num_rows($result);
            $num_col = mysql_num_fields($result);
            for($icol=0; $icol < $num_col; $icol++)
                  $recordset[0][$icol]=mysql_field_name($result, $icol);
            echo("</tr>");
            for($irow=1; $irow <= $num_row; $irow++){
                  $row = mysql_fetch_array($result);
                  for($icol=0; $icol < $num_col; $icol++)
                        $recordset[$irow][$icol]=$row[$icol];
            }
            mysql_close($con);
            return($recordset);
```

```
	}

	function tabView($recordset, $num_row, $num_col){
		echo("<table border=\"1\">");
		for($irow=0; $irow <= $num_row; $irow++)  {
			echo("<tr>");
			for($icol=0; $icol < $num_col; $icol++)
				echo '<td>' . $recordset[$irow][$icol] . '</td>';
			echo("</tr>");
		}
		echo("</table>");
	}
?>
<!DOCTYPE html>

<html>
<head>
	<title>RUBRICA 2015</title>
	<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
    <h1>CERCA TEL</h1>
	<form action="CercaTel.php" method="post">
		Cerca Telefono per Cognome
<?php
		include("conn_db.php");
		$query = "SELECT DISTINCT Cognome FROM utenti ORDER BY Cognome";
		$recordset=estraiRecord($num_row, $num_col, $query);
		echo '<select name="cognome">';
		for($i=1; $i<=$num_row; $i++){
			echo '<option value="' .
			   $recordset[$i][0] . '">' . $recordset[$i][0] . '</option>';
		}
		echo '</select>';
?>
		<br />
		<input type="submit" value="CERCA" />
	</form>

<?php
	if(isset($_POST["cognome"])){
		$cognome = $_POST["cognome"];
```

```
            $query = "SELECT * FROM utenti WHERE Cognome = '$cognome'
                      ORDER BY Cognome, Nome";
            $recordset=estraiRecord($num_row, $num_col, $query);
            tabView($recordset, $num_row, $num_col);
      }
?>
```

LE VARIABILI GLOBALI $_POST E $_GET

Le variabili di tipo POST e di tipo GET sono, a tutti gli effetti, quelle di maggiore importanza nell'ambito di siti dinamici: esse consentono di ricevere dei dati da parte degli utenti attraverso un form HTML o attraverso l'URL e, a seconda di ciò che l'utente avrà immesso, si potrà interagire con esso.
Tale interazione è determinata soprattutto dalla **caratteristica fondamentale delle variabili POST e GET di essere valorizzate direttamente (o indirettamente) dall'utente**: i contenuti e la risposta del sito cambieranno a seconda della valorizzazione che gli utenti faranno di tali variabili. Tuttavia questa stessa caratteristica (la valorizzazione gestibile dall'utente) rende tali variabili particolarmente *delicate da utilizzare* e *potenzialmente pericolose* qualora non siano utilizzate opportunamente.

Premetto fin da subito che un loro cattivo e/o superficiale utilizzo può far incorrere in gravi rischi e, pertanto, richiedono una maggiore delicatezza nell'impiego. In questo punto della guida non tratteremo alcuni aspetti legati alla *validazione dei dati*, ma occorre già da adesso tenerli in considerazione.

Esse, inoltre, si caratterizzano per essere della variabili globali, o meglio, degli array globali: ciò sta a significare che esse sono variabili che **esistono in qualsiasi pagina php**, ma magari essere vuote.

L'ulteriore caratteristica di tali variabili è, inoltre, **la possibilità di essere trasmesse da una pagina all'altra**. Per capire questa seconda caratteristica supponiamo di avere due pagine distinte con i seguenti codici:

```
<?php
$var = "valore della stringa";
?>
```

```
<?php
echo $var;
?>
```

La variabile $var è valorizzata nella prima pagina, mentre nella seconda pagina quella stessa variabile non essendo stata valorizzata risulterà non esistente. Le variabili POST e GET costituiscono uno dei metodi per trasmettere variabili da una pagina web ad un'altra.

Quindi, ricapitolando, le caratteristiche della variabili POST e GET sono:

1. la valorizzazione fatta direttamente o indirettamente dall'utente;
2. la loro esistenza in tutte le pagine php, anche se vuote;
3. la possibilità di essere trasmesse da una pagina ad un'altra.

A questo punto vediamone separatamente tali caratteristiche dal punto di vista pratico iniziando prima con le variabili GET e poi passando alle variabili POST.

VARIABILI GET

Le variabili GET sono valorizzate attraverso l'url della pagina (tecnicamente di parla di **QueryString**) attraverso una sintassi di questo tipo:

www.sito.it/directory/page.php?var=1

Esaminiamo questo url: stiamo visitando il sito "*www.sito.it*" e stiamo visualizzando la pagina "*page.php*" contenuta all'interno della cartella "*directory*". All'interno dell'URL troviamo anche una particolare sintassi:

"**?var=1**".

In questo modo stiamo inviando la variabile GET chaimata "var" e la stiamo valorizzando con 1.

Vediamo come si inviano tali variabili e il modo in cui si fa a riceverle. Facciamo una prova creando le seguenti pagine web "invia.php" e "ricevi.php" (in questo esempio mettetele per semplicità in una stessa cartella):

invia.php

```
<html>
<body>
<a href="ricevi.php?var=1">Invia variabile get 1</a>
<a href="ricevi.php?var=2">Invia variabile get 2</a>
</body>
<html>
```

rivevi.php

```
<?php
$variabile_get = $_GET['var'];

echo "La variabile get che abbiamo inviato è ".$variabile_get;
?>
```

Puntiamo il browser sulla pagina "invia.php" che è una semplice pagina web con due link entrambi alla pagina "ricevi.php" ma con due variabili GET differenti: in un caso "?var=1" nell'altro "?var=2".

La pagina "ricevi.php" modificherà i suoi contenuti in base alla variabile GET che riceverà, cioè a seconda del link scelto dall'utente, ottenendo così un primo (e semplice) esempio di dinamicità.

La valorizzazione della variabile GET in questo caso è stata fatta (indirettamente) dall'utente che ha scelto di cliccare su un link piuttosto che su un altro.

A questo punto vediamo il modo in cui possiamo semplicemente inviare una variabile php presente in una pagina ad un'altra pagina con metodo GET. Modifichiamo il file "invia.php" in questo modo:

invia.php

```
<html>
<body>
<?php
$primo=1;
$secondo=2;

echo "<a href=ricevi.php?var=".$primo.">";
echo "<a href=ricevi.php?var=".$secondo.">";
?>
</body>
<html>
```

In questo modo le variabili $primo e $secondo sono state trasmesse dalla pagina "invia.php" alla pagina "ricevi.php".

VARIABILI POST

Le variabili di tipo POST sono valorizzate attraverso i form.

In qualsiasi sito di tipo dinamico non mancherà mai un modulo per l'invio si dati a vario scopo: la loro costruzione non fa parte della presente guida in quanto questi sono realizzati con linguaggio HTML ma mi limiterò solo a fare alcuni appunti.

Il tag <form> è caratterizzato dalla seguente sintassi di base:

<form action="riceve.php" method="post">

Esso deve contenere nell'*action* la pagina php a cui inviare i dati e nel *method* il metodo utilizzato per l'invio, che potrà essere per l'appunto, get o post. Nel caso in cui l'action è vuota i dati saranno

inviati alla pagina stessa in cui è situato il form. Per quel che riguarda il "method" pur potendo essere di due tipi generalmente è preferibile utilizzare il metodo post.

Il tag form potrà contenere anche l'*enctype* (che assumerà rilevanza soprattutto se si devono inviare file), il *name* e l'*id*(utile a supporto dei ccss e/o javascript) e il target (nel caso si voglia inviare i dati ad una finestra separata, nel caso di utilizzo di frame o di iframe).

Dopo tale tag vi saranno i campi di cui sarà costituito il modulo (text, password, select e option, checkbox, radio button) ognuno dei quali avrà un "**name**", che identificherà la variabile inviata. Ad esempio un tag input si tipo text avrà una sintassi di questo tipo:

<input type="text" name="campo" />

Infine vi sarà l'input di **type submit** per inviare i dati alla pagina specificata nell'action.

Per una trattazione maggiormente specifica sulla costruzione dei form vi invito a consultare una specifica guida trattante il linguaggio HTML.

Ai fini della presente guida consideriamo un form molto semplice e basilare messo all'interno di una pagina che chiameremo invia.php:

```
<form action="riceve.php" method="post">
<p>Inserisci un valore</p>
<input type="text" name="valore" />
<input type="submit" value="Invia" />
</form>
```

Questo form al clic del tasto "Invia" manderà con metodo "post" il valore inserito nel campo a cui abbiamo attribuito il name "valore" al file php chiamato "riceve.php".

Quest'ultimo per recepire il dato che gli è stato inviato avrà la seguente sintassi:

```
<?php
```

```
$variabile = $_POST['valore'];

echo $variabile;
```

1. ?>

Si può quindi capire che il contenuto della pagina "riceve.php" è del tutto controllabile dall'utente che potrà inserire nel campo del form qualsiasi valore lui desideri.

VARIABILI GET E POST

I linguaggi di scripting lato server generano contenuti (dinamicamente) in risposta a richieste dell'utente.

Questa interazione è realizzabile attraverso due tecniche differenti, **GET** e **POST** che fanno uso rispettivamente delle variabili (globali) **$_GET** e **$_POST**.

Col metodo GET i dati vengono passati in chiaro all'interno dell'URL il quale si presenterà nella forma:

elabora.php?a=2&b=3&c=4

Sebbene il metodo GET possa essere utilizzato anche nei form si preferisce in questi il metodo POST in quanto i dati vengono passati non in chiaro ma criptati.

INVIO DATI CON IL METODO POST VIA E-MAIL

FORM.HTML

```
<!doctype html>
```

```
<html lang="it">
<head>
<meta charset="utf-8" />
<title>Form Registrazione</title>
</head>
<body>
<strong>Dati di registrazione</strong>
<form action="elaboraForm.php" method="post" name="FormRegistrazione">
<strong>Nome:</strong><br />
<input type="text" name="nome" /><br />
<strong>Cognome:</strong><br />
<input type="text" name="cognome" /><br />
<strong>Email:</strong><br />
<input type="text" name="email" /><br />
<strong>Username:</strong><br />
<input type="text" name="username"  /><br />
<strong>Password:</strong><br />
<input type="password" name="password" /><br /><br />

<strong>Utente</strong><br />
<input type="radio" name="utente" value="alunno" />Alunno<br />
<input type="radio" name="utente" value="docente" />Docente<br /><br />

<strong>Come sei venuto a conoscenza di questo sito?</p>
<input  type="checkbox"  name="vetConoscenze[]"  value="web"  checked="checked"
/>Web<br />
<input type="checkbox" name="vetConoscenze[]" value="amico" />Amico<br />
<input type="checkbox" name="vetConoscenze[]" value="rivista" />Rivista<br /><br
/><br />

<strong>Commento</strong><br />
 <textarea rows="4" cols="50" name="commento">
</textarea>
<br /><br />

<strong>Numeri da inviare:</strong><br />
<input type="text" name="numeri" /><br />

<br /><br />
<input type="submit" value="Registrati" />
<input type="reset" value="Annulla" />
</form>
```

```
</body>
</html>
```

ELABORA.PHP

```
<?php
	header('Content-type: text/html;charset=utf-8');
	$destinatario="daniele.corti@liceotosi.va.it";
	$mittente=$_POST['email'];
	$nome=$_POST['nome'];
	$cognome=$_POST['cognome'];
	$username=$_POST['username'];
	$password=$_POST['password'];
	$utente=$_POST['utente'];
	$vetConoscenze=$_POST['vetConoscenze'];
	$commento=$_POST['commento'];
	$numeri=$_POST['numeri'];
	$vetNumeri=explode(" ",$numeri);
	$contaNum=count($vetNumeri);
	for($i=0; $i<$contaNum; $i++){
		echo $vetNumeri[$i] . " ";
	}
	$media=0;
	for($i=0; $i<$contaNum; $i++){
		$media=$vetNumeri[$i]+$media;
	}
	$media=$media/$contaNum;
	echo "Media: " . $media . "<br />";
	if(!$nome){
		echo "Inserisci il nome";
	}
	else if(!$cognome){
		echo "Inserisci il cognome";
	}
	else if(!$mittente){
		echo "Inserisci la e-mail";
	}
	else if(!$username){
		echo "Inserisci la username";
	}
```

```
	else if(!$password){
		echo "Inserisci la password";
	}
	else if(!$utente){
		echo "Scegli che tipo di utente sei";
	}
	else if(!$vetConoscenze){
		echo "Almeno una selezione dal checkbox Conoscenze.";
	}
	else{
		/*foreach($vetConoscenze as $conosc) {
			echo $conosc . " ";
		}
		*/
		$num=count($vetConoscenze);
		$conosc="";
		for($i=0; $i<$num; $i++){
			$conosc=$conosc . ", " . $vetConoscenze[$i];
		}

		$intestazione .= "From: {$_POST['email']} <$mittente> \r\n";
		$oggetto="Invio email da: " . $_POST['cognome'] . ' ' .
$_POST['nome'];
		$messaggio="Questa email e' stata inviata da ";
		$messaggio .= $cognome . " " . $nome . ".\nEmail: " . $mittente;
		$messaggio .= "\nUsername: " . $username . "\nPassword: " .
$password;
		$messaggio .= "\nUtente: " . $utente . "\nSono venuto a conoscenza
del sito attraverso: ";
		$messaggio .= $conosc . "\nCommento: " . $commento;
		if(@mail($destinatario, $oggetto , $messaggio, $intestazione)){
			echo "La e-mail è stata inviata con successo!";
		}
		else{
			echo "Si è verificato un errore!";
		}
	}
?>
```

INSERIRE UN NUOVO RECORD NEL DB DA FORM

AGGIORNARE UNO O PIU' RECORD

ELIMINA UNO O PIU' RECORD