

Pagine dinamiche PHP e connessioni a database MySQL

4c, febbraio 2012

Cominciamo con un esempio.

Supponiamo di avere un database MySQL, con una tabella di nome **tabella_esempio** che contiene i campi (**id**, **nome**, **cognome**, **password**).

id	nome	cognome	password
1	pippo	baudo	ccccqpxzczczxc
11	paolino	paperino	ccccf
69	Marco	Tomazzoli	nonlaso

Ora supponiamo di dover utilizzare questa tabella all'interno di un nostro sito web. Dovrà quindi essere possibile, sempre via web, accedere al database in modo da

1. visualizzare il contenuto della tabella
2. inserire nuovi dati
3. modificare i dati esistenti o rimuoverli
- 4.

Presupposti

Ricordiamo che il database è gestito da un apposito server, a cui avremo accesso con appropriati credenziali (username e password). Dal nostro script PHP dovremo quindi **aprire** una connessione a questo database specificando al server MySQL username e password.. Potremo quindi selezionare il database sul quale vogliamo lavorare (specificandone il nome), per poi eseguire tutti i comandi SQL (SELECT, INSERT etc etc) sul nostro database.

Per connetterci al database usando PHP dovremo usare la funzione **mysql_connect** a cui dovremo passare i seguenti parametri, precedentemente definiti:

```
$mydbconn=mysql_connect($mysql_host, $mysql_user, $mysql_password);
```

nel nostro caso useremo i seguenti parametri:

```
$mysql_host = "mysql3.000webhost.com"; // nome del server su cui gira MySQL  
$mysql_user = "a4882837_pippo"; // nome utente MySQL  
$mysql_password = "12345qwerty"; // password MySQL
```

Dovremo quindi selezionare anche il database sul quale vogliamo lavorare, con l'apposita funzione `mysql_select_db`
mysql_select_db(\$mysql_database);

naturamlemnet dopo aver impostato opportunamente la variabile corrispondente al nome del database

```
$mysql_database = "a4882837_pippo";
```

Un po' di SQL

Supponiamo di avere una tabella con le caratteristiche menzionate all'inizio. Vediamo come possiamo, usando il linguaggio SQL,

- **inserire** un nuovo dato:

```
INSERT INTO tabella_esempio ( nome, cognome) VALUES ("pippo", "baudo");
```

nel caso vengano specificati i valori di tutti i campi della tabella, si possono specificare semplicemente i valori senza elencare tutti i campi, es:

```
INSERT INTO tabella_esempio VALUES (118,"pippo", "baudo", "la mia password");
```
- **rimuovere** uno (o più record) da una tabella:

```
DELETE FROM tabella_esempio WHERE nome="paolino" AND cognome="paperino";
```
- **modificare** uno(o più) dati esistenti:

```
UPDATE tabella_esempio SET nome="topolino", password="pluto" WHERE id="11";
```
- **selezionare** tutti i dati presenti in una tabella

```
SELECT * FROM tabella_esempio;
```
- **selezionare** solo alcune colonne di una tabella

```
SELECT cognome, nome FROM tabella_esempio;
```
- **selezionare** solo alcuni record (righe) dalla tabella

```
SELECT cognome, nome FROM tabella_esempio WHERE cognome="paperino" OR id < 100;
```

Visualizzazione di una tabella

Nella nostra pagina PHP, dovremo innanzitutto: aprire la connessione al server MySQL, selezionare il database sul quale vogliamo lavorare, quindi eseguire la richiesta SQL per selezionare tutti i dati presenti nella tabella, e rappresentarli tutti formattando una tabella HTML. Ma procediamo con ordine. La parte iniziale corrisponderà a qualcosa del tipo:

```
<?php
    $mysql_host = "mysql3.000webhost.com"; // nome del server su cui gira MySQL
    $mysql_user = "a4882837_pippo"; // nome utente MySQL
    $mysql_password = "12345qwerty"; // password MySQL
    $mysql_database = "a4882837_pippo"; //nome del database
    $mydbconn=mysql_connect($mysql_host, $mysql_user, $mysql_password);
    mysql_select_db($mysql_database);
?>
<html>
  <body>
    <table>
  <?php
    //qua inseriremo tutto il codice (sotto riportato) che estrae i dati dal database
?>
  </table>
</body>
</html>
```

Ora dovremo selezionare tutti i dati della tabella in questione; lo faremo usando la funzione **mysql_query**

```
$query = "SELECT * FROM tabella_eseempio";
$result=mysql_query($query);
```

Possiamo controllare quanto risultati(quante righe) otteniamo dalla nostra richiesta con la funzione **mysql_numrows**

```
$num=mysql_numrows($result);
```

e quindi andare a selezionare una riga alla volta usando la funzione **mysql_fetch_row** e dalla quale otteniamo un array (\$row) con un valore per ogni campo (colonna) della tabella. Usando il comodissimo ciclo foreach del PHP possiamo quindi riassumere questa parte nel seguente codice:

```
for($i=0; $i<$num; $i++){ // da 0 al numero dei risultati
    echo "<tr>\n";
    $row=mysql_fetch_row($result); // seleziono la riga successiva del risultato
    foreach($row as $val) { // per ogni elemento $val dell'array $row
        echo "<td>$val</td>\n"; // stampo la cella html
    };
    echo "</tr>\n";
};
```

Volendo è possibile anche visualizzare l'intestazione della tabella, elencando i nomi delle colonne. Anche questo si può fare estraendo i dati dal database, usando la funzione **mysql_field_name** e scrivendo il codice PHP come di seguito:

```
echo "<tr>\n";
while($label=@mysql_field_name($result, $i++){
    echo "<th>$label</th>\n";
};
echo "</tr>\n";
```

Compito

Crea una tabella a piacere al link http://sql3.000webhost.com/phpMyAdmin/?db=a4882837_pippo usando l'username **a4882837_pippo** e la password **12345qwerty**

Allo stesso link potrai inserire anche alcuni dati nella tabella.

Crea quindi una pagina PHP per visualizzare i dati contenuti in questa tabella. La pagina php potrai caricarla e testarla al link <http://russe.net84.net/simple.php>

Per inserire un nuovo dato in un database tramite interfaccia web, oppure per modificare o eliminare dati nel database, bisogna preparare un modulo che permetta l'inserimento o la selezione di questi dati; tali dati dovranno essere spediti ad un apposito script PHP che eseguirà le opportune operazioni sul database. Si rende quindi necessario strutturare il sito per l'immissione di questi dati, quindi per l'invio di tali parametri ad apposito script PHP che li elaborerà.

Il passaggio di parametri da una pagina web al PHP si svolge quindi in due fasi:

1. La prima, dove bisogna preparare una pagina HTML con un modulo (form) per il riempimento dei campi da parte dell'utente.
2. La seconda, dove una pagina PHP deve ricevere questi dati ed elaborarli, salvarli etc.

Nella prima parte all'interno di una pagina HTML sarà necessario usare il tag **form** per racchiudere tutti gli elementi html che serviranno per inserire dati (pulsanti, caselle di selezione, menù a tendina, caselle di inserimento testo, modulo per l'upload dei files etc).

```
<form action="elabora.php" method="GET">
...
</form>
```

Come parametri di questo tag, è necessario specificare la pagina a cui verranno inviati i dati (in questo caso "elabora.php") ed il metodo con cui inviare i dati (GET oppure POST). All'interno di questo blocco HTML sarà quindi possibile inserire, oltre ai normalissimi elementi HTML già noti, anche gli opportuni tag per l'inserimento dei dati. Ad ognuno di questi dovrà essere necessariamente assegnato un nome da associare al valore inserito, per riuscire ad elaborare i vari dati nello script PHP.

Questi sono un paio di esempi dei principali elementi di input:

- casella per l'inserimento di testo

```
<input type="text" name="utente" value="Amilcare" size="50">
```
- casella per l'inserimento di una password

```
<input type="password" name="password" size="50">
```
- casella per l'inserimento di un blocco di testo

```
<textarea name="comunicazione" rows="12" cols="60">
    Inserisci qui il corpo del testo
</textarea>
```
- pulsanti a selezione unica

```
<input type="radio" name="biglietto" value="giornaliero" >Giornaliero
<input type="radio" name="biglietto" value="settimanale" checked >Settimanale
```
- pulsanti a selezione multipla (la parentesi quadra serve a far capire al PHP che la variabile servizi sarà un array formato da più elementi)

```
<input type="checkbox" name="servizi[]" value="ristorante" >Carrozza Ristorante
<input type="checkbox" name="servizi[]" value="caffè" checked >Caffè
```
- menu a tendina (l'opzione size serve solo per visualizzare più elementi assieme, l'opzione multiple per dar la possibilità di selezionare più di un elemento; nel caso di scelte multiple, vanno messe le parentesi quadre dopo il nome del select per far capire al PHP che la variabile giorni è un array):

```
<select name="giorni[]" size="4" multiple>
<optgroup label="Feriali">
  <option value="1" >Lunedì</option>
  <option value="2" selected >Martedì</option>
  <option value="3" selected >Mercoledì</option>
  <option value="4" >Giovedì</option>
  <option value="5" >Venerdì</option>
</optgroup>
<optgroup label="Festivi">
  <option value="6" >Sabato</option>
```

```

    <option value="7" >Domenica</option>
</optgroup>
</select>

```

- pulsante di reset
- pulsante per l'invio dei dati

Nella seconda parte lo script PHP di cui all'indirizzo specificato nel parametro **action** del form HTML (**elabora.php** nell'esempio sopra riportato), dovrà contenere del codice opportuno per elaborare tali dati.

Tutte le informazioni inviate dal form HTML, sono salvate nell'array predefinito **\$_REQUEST** (oppure, in alternativa, negli array **\$_POST** o **\$_GET** a seconda di quale metodo è stato specificato come parametro nel form HTML).

Pertanto per accedere alle informazioni specificate nel form precedente, basterà semplicemente accedere all'elemento dell'array

```
$_REQUEST["utente"]
```

per leggere il nome dell'utente inserito nel form HTML. Stessa cosa dicasi per tutti gli altri input di tipo **text**, **password**, **radio**, **submit**, **textarea** e **select** singoli.

Gli elementi di tipo **checkbox** e **select multipli** corrisponderanno invece ad un array, ed i singoli elementi saranno accessibili nel seguente modo:

```

foreach($_REQUEST["servizi"] as $val){
    // la variabile $val contiene il valore del singolo elemento
}

```

Inserimento di un dato nel database

Torniamo all'esempio della volta scorsa, assumendo che esista nel database MySQL una tabella di nome **tabella_esempio** che contiene i campi (**id**, **nome**, **cognome**, **password**).

Innanzitutto dobbiamo creare una pagina HTML (supponiamo **inserisci1.html**) che predisponga il form per l'inserimento. Il cuore della pagina potrebbe essere il seguente:

```

<form action="inserisci2.php">
  Inserisci l'id      <input type="text" name="id"><br>
  Inserisci il nome   <input type="text" name="nome"><br>
  Inserisci il cognome <input type="text" name="cognome"><br>
  Inserisci la password <input type="password" name="password"><br>
  <input type="submit" value="Invia i dati">
</form>

```

Ora bisogna creare una pagina **inserisci2.php** che elabori le precedenti informazioni inserendole nel database. La chiamata SQL da eseguire sarà la seguente:

```
INSERT INTO tabella_esempio VALUES ("??", "??", "??", "??");
```

Al posto dei ?? andrà chiaramente l'opportuno valore passato al PHP dalla precedente pagina HTML.

Ricordando che il punto . in php funziona come operatore di concatenamento delle stringhe, vediamo come costruire la stringa \$query corrispondente al comando SQL da eseguire (attenzione agli apici doppi e singoli!!!):

```

$query = "INSERT INTO tabella_esempio VALUES ('$_REQUEST[id]', '$_REQUEST[nome]',
'$_REQUEST[cognome]', '$_REQUEST[password]'); ";

```

Dopo aver aperto la connessione al database come già visto per visualizzare una tabella, basta semplicemente eseguire

la richiesta sul database MySQL.

```
$result=mysql_query($query);
echo "<p>L'inserimento dovrebbe essere avvenuto...</p>
      <p>Torna alla pagina di visualizzazione della tabella per controllare</p>";
```

Cancellazione di un dato dal database

Possiamo anche predisporre una pagina per l'eliminazione di uno o più dati dal database. Le modalità sono svariate, noi proviamo a eliminare i dati andando a specificare la chiave primaria dell'elemento da eliminare.

Possiamo preparare una pagina HTML contenente il seguente form:

```
<form action="elimina2.php">
  Inserisci l'id dell'elemento da cancellare <input type="text" name="id"><br>
  <input type="submit" value="elimina record">
</form>
```

e quindi preparare anche l'opportuna pagina **elimina2.php** che provvede all'esecuzione della query SQL. In modo simile al precedente esempio si avrà una struttura di questo genere (attenzione agli apici singoli e doppi!!):

```
$query ="DELETE FROM tabella_eseempio  where id='$_REQUEST[id]' " ;
$result=mysql_query($query);
echo "<p>L'eliminazione dovrebbe essere avvenuta...</p>
      <p>Torna alla pagina di visualizzazione della tabella per controllare</p>";
```

Modifica di un dato nel database

Innanzitutto predisponiamo una pagina (**pagina1.html**) per andare a selezionare il dato da modificare, del tutto simile a quella usata per l'eliminazione del dato:

```
<form action="modifica2.php">
  Inserisci l'id dell'elemento da modificare <input type="text" name="id"><br>
  <input name="azione" type="submit" value="modifica record">
</form>
```

Quindi dobbiamo preparare una pagina del tutto simile a quella usata per l'inserimento di un dato, ma con il form precompilato con i dati già presenti nel database. Dobbiamo quindi selezionare dal database l'elemento corrispondente all'id (ovvero a \$_REQUEST["id"]) specificato, e predisporre il form. Dopo i soliti convenevoli per aprire la connessione, scriveremo:

```
<?php
  $query = "SELECT * FROM tabella_eseempio WHERE id='$_REQUEST[id]' ";
  $result=mysql_query($query);
  $row=mysql_fetch_array($result,MYSQL_ASSOC);
?>
<form action="modifica3.php">
  <input type="hydden" name="idOLD" value=<?=$ _REQUEST["id"]?> ><br>
  id      <input type="text" name="id" value=<?=$ _REQUEST["id"]?> ><br>
  nome    <input type="text" name="nome" value=<?=$ _REQUEST["nome"]?> ><br>
  cognome <input type="text" name="cognome" value=<?=$ _REQUEST["cognome"]?> ><br>
  password <input type="password" name="password" value=<?=$ _REQUEST["password"]?> ><br>
  <input type="submit" value="Salva">
</form>
```

Come avrai notato abbiamo messo anche il campo idOLD come variabile nascosta, per ricordarci quale record

dovevamo modificare nel caso si voglia aggiornare anche l'id.

La pagina **modifica3.php** dovrà quindi eseguire la query di salvataggio dei dati nel database, ovvero (attenzione agli apici singoli e doppi!!):

```
$query = "UPDATE tabella_esempio SET id='$_REQUEST[id]', nome='$_REQUEST[nome]',  
cognome='$_REQUEST[cognome]', password='$_REQUEST[password]' where id='$_REQUEST[idOLD]' ";  
$result=mysql_query($query);  
echo "<p>L'aggiornamento dovrebbe essere avvenuto...</p>  
<p>Torna alla pagina di visualizzazione della tabella per controllare</p>";
```

Passaggi di parametri col metodo GET

Considera uno degli esempi precedenti e, usando il metodo GET nel form delle varie pagine HTML, fai attenzione a come la relativa pagina PHP viene richiamata nella barra degli indirizzi quando fai il submit del form. Noterai che nella barra degli indirizzi compare, dopo il nome della pagina richiesta, un punto di domanda seguito dal nome della variabile coi rispettivi valori. Vediamo l'esempio:

<http://www.miosito.it/elabora2.php?nome=Amilcare&password=pippo&codicefiscale=ASD&invia=spedisce>

In questo caso viene richiamata la pagine elabora2.php mandandogli anche le 4 variabili **nome**, **password**, **codicefiscale** e **invia**. L'elenco delle variabili comincia con un punto di domanda ? subito dopo il nome della pagina. Le variabili si susseguono separate da una & nel formato, **nome_variabile=valore**.

In questo modo tutte le informazioni vengono passate direttamente assieme all'indirizzo della pagina stessa. Questo dà la possibilità di creare in alcune pagine HTML dei link che svolgono la stessa funzione di una pagina con un form, che permettono di specificare uno o più parametri da inviare alla pagina.

Vediamone ora l'utilità.

Consideriamo l'esempio della modifica (o dell'eliminazione) di un dato dal database. Avevamo avuto la necessità di creare una prima pagina HTML per specificare l'id del record da modificare. Supponendo di aver specificato un **id=75**, la richiesta alla pagina risulta essere qualcosa del tipo:

<http://www.miosito.it/modifica2.php?id=75&azione=modifica+record>

È evidente che, andando a ripescare la pagina che serviva per la visualizzazione della tabella, può risultare molto più comodo mettere a fianco di ogni record visualizzato un link "modifica" che punti ad un indirizzo come il precedente (specificando l'appropriato valore dell'id, chiaramente!).

Stessa cosa si può chiaramente fare anche per l'eliminazione di un record, ovvero mettere a fianco di ogni elemento della tabella un link

<http://www.miosito.it/elimina2.php?id=75&azione=modifica+record>

È più che evidente che dare la possibilità di modifica e eliminazione dei direttamente dalla tabella che visualizza l'elenco dei dati può essere molto più comodo ed intuitivo dell'approccio visto precedentemente.

id	nome	cognome	password		
1	pippo	baudo	topolino	Rimuovi	Modifica
11	Geronimo	Odorizzi	nonlaso	Rimuovi	Modifica
69	Marco	Tomazzoli	sandalino<3	Rimuovi	Modifica

Compito

Se non l'hai già fatto, crea una tabella a piacere al link http://sql3.000webhost.com/phpMyAdmin/?db=a4882837_pippo usando l'username **a4882837_pippo** e la password **12345qwerty**

Allo stesso link potrai inserire anche alcuni dati nella tabella.

L'altra volta avevi creato una pagina PHP per visualizzare i dati contenuti in questa tabella. Ora prova a creare quelle necessarie per l'inserimento, la modifica e l'eliminazione dei dati. La pagina php potrai caricarla e testarla al link <http://russel.net84.net/simple.php>

Esempi e suggerimenti li trovi come sempre anche su Claroline 1.9.