Applicazione client-server in PHP con database MySQL

Emanuele Scapin[1]

[1]Liceo Statale F.Corradini, via Milano 1, 36016 Thiene

30/05/2012

1 Premessa

Dopo la trattazione di argomenti teorici di progettazione di Basi di Dati (database)[1] si vuole proporre la creazione di una semplice applicazione *client-server*[2]. Si farà ricorso a codice *SQL*[3] per costruire delle query da applicare a un database *MySQL*[4], le interrogazioni verranno implementate grazie a codice in linguaggio *PHP*[5].

2 Requisiti: conoscenze dello studente

Lo studente deve necessariamente avere acquisto competenze in merito ai seguenti argomenti teorici precedentemente trattati:

[1]Il termine database o base di dati indica un insieme di archivi collegati secondo un particolare modello logico (attualmente il modello relazionale) e in modo tale da consentire la gestione dei dati.

[2]Un'applicazione client-server è un tipo di applicazione di rete nel quale un computer client (cliente) istanzia l'interfaccia utente di un'applicazione la quale si connette ad una server application (applicazione server) oppure a un sistema di database.

[3]SQL (Structured Query Language) è un linguaggio di interrogazione per database.

[4]Un database relazionale tra i più popolari, la versione base è gratuita.

[5]PHP (acronimo di Hypertext Preprocessor), è un linguaggio di scripting interpretato, con licenza open source e libera, originariamente concepito per la programmazione Web ovvero la realizzazione di pagine web dinamiche.

- progettazione concettuale e schema entità/relazioni (ER);
- progettazione logica e schema relazionale, vincoli relazionali e ottimizzazione (forme normali);
- istruzioni del linguaggio Sql.

Deve inoltre ricordare i seguenti argomenti trattati gli anni precedenti:

- linguaggio Html;
- programmazione con linguaggio C++.

3 Requisiti: hardware e software

Per implementare una applicazione *client-server* si deve utilizzare una rete locale (LAN) dove ogni studente avrà a disposizione una postazione di lavoro. Per utilizzare efficacemente le conoscenze sulle reti si deve evitare di installare il database MySQL[6] e il web server Apache[7] con interprete Php in ogni singola macchina ma invece si preferirà utilizzare un server condiviso in cui fare le installazioni. Sul server viene prima installato il database MySQL con le opportune configurazioni, successivamente viene installato Apache, una volta installato Apache si installa l'interprete di Php[8] facendo attenzione alla manipolazione dei file di configurazione in modo da rendere visibile l'interprete al web server.
Righe da inserire nel file Httpd.conf di Apache per permettere l'invocazione dell'interprete Php.

```
# Start Php integration
LoadModule php5_module "c:/programmi/php/php5apache2_2.dll"
AddType application/x-httpd-php .php
PhpIniDir "C:/programmi/php"
#End Php integration
```

Ora Apache deve pubblicare sia file Html che file Php.

[6] Reperibile sul sito www.mysql.it.
[7] Reperibile sul sito httpd.apache.org.
[8] Reperibile sul sito www.php.net.

```
<IfModule dir_module>
    DirectoryIndex index.html index.php
</IfModule>
```

Il web server fa uso di una cartella dove verranno pubblicate le pagine web in Html e in Php, questa cartella deve essere condivisa in rete da tutte le postazioni di lovoro. Infine, per operare agevolmente con il database si installa PhpMyAdmin[9], questa applicazione fa uso del web server e del linguaggio Php e offre una interfaccia grafica per la gestione del database MySQL. Tutti i pacchetti software utilizzati e installati sono reperibili gratuitamente in rete, infatti sono concessi con licenza GPL[10].

4 Applicazioni

Di seguito si presentano delle applicazioni che possano guidare lo studente all'uso di Php per interrogare e manipolare un database MySQL tenendo presente un approccio graduale a passi successivi di crescita delle competenze, come enunciato precedentemente.

4.1 Php, un primo esempio

Per introdurre il linguaggio Php vale la pena proporre un piccolo esempio dove gli studenti potranno subito verificare analogie e differenze con il linguaggio C++ già di loro conoscenza.

```
<?php
$i = 0;
echo('<table border="1" cellpadding="10">');
while($i < 10) {
    echo("<tr><td>riga $i</td><td>altra cella</td></tr>");
    $i++;
}
echo("</table>");
?>
```

il risultato sarà una semplce tabella come nell'immagine seguente.

[9] Reperibile sul sito http://www.phpmyadmin.net.

[10] La GNU General Public License, comunemente indicata con l'acronimo GNU GPL o semplicemente GPL, è una licenza per software libero.

riga 0	altra cella
riga 1	altra cella
riga 2	altra cella
riga 3	altra cella
riga 4	altra cella
riga 5	altra cella
riga 6	altra cella
riga 7	altra cella
riga 8	altra cella
riga 9	altra cella

4.2 MySQL, la prima tabella

Necessariamente per operare su un database si deve costruire uno schema con almeno una tabella con qualche campo su cui scrivere dei dati e leggerli. Se la tabella viene creata con l'applicazione MyPhpAdmin la si potrà costruire tramite interfaccia grafica ottenendo il seguente risultato.

	#	Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra	Azione
☐	1	id	int(10)		UNSIGNED	No	*Nessuno*	AUTO_INCREMENT	Più
☐	2	nome	varchar(50)	latin1_swedish_ci		No	*Nessuno*		Più
☐	3	cognome	varchar(50)	latin1_swedish_ci		No	*Nessuno*		Più
☐	4	indirizzo	varchar(50)	latin1_swedish_ci		Sì	*NULL*		Più
☐	5	comune	varchar(30)	latin1_swedish_ci		Sì	*NULL*		Più
☐	6	cap	varchar(5)	latin1_swedish_ci		Sì	*NULL*		Più

Lo script Sql equivalente per la creazione della tabella è il seguente

```
CREATE TABLE `studente` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `nome` varchar(50) NOT NULL,
  `cognome` varchar(50) NOT NULL,
  `indirizzo` varchar(50) default NULL,
  `comune` varchar(30) default NULL,
  `cap` varchar(5) default NULL,
  PRIMARY KEY (`id`)
)
```

4.3 Html e Php, il tag form

Il file index.html o index.php (anche se in realtà non contiene codice Php) sarà del tipo

```
<html>
    <head>
        <title></title>
    </head>
    <body>
        <form action="form_action.php" method="get">
          Nome: <input type="text" name="fnome" /><br/>
          Cognome: <input type="text" name="fcognome" /><br/>
          Indirizzo <input type="text" name="findirizzo" /><br/>
          Comune <input type="text" name="fcomune" /><br/>
          Cap <input type="text" name="fcap" /><br/>
          <input type="submit" value="Submit" />
        </form>
    </body>
</html>
```

dove è presente una form con degli input di tipo *text* per l'inserimento dei dati e uno di tipo *submit* per invocare l'avvio della action. L'invocazione della *action*, digitando il tasto *Submit*, produrrà la seguente richiesta della pagina

```
form_action.php?fnome=luca&fcognome=rossi&
findirizzo=via+verdi&fcomune=thiene&fcap=36016
```

che passerà con metodo *get* i dati alla pagina Php che li dovrà gestire, il cui nome deve coincidere con quello inserito nella *action* della form, il passaggio dei dati avviene secondo il protocollo Http[11],

```
<?php
$nome = $_GET['fnome'];
$cognome = $_GET['fcognome'];
$indirizzo = $_GET['findirizzo'];
$comune = $_GET['fcomune'];
$cap = $_GET['fcap'];

echo('<h3>nome: '.$nome.'</h3><br/>');
echo('<h3>cognome: '.$cognome.'</h3><br/>');
echo('<h3>indirizzo: '.$indirizzo.'</h3><br/>');
echo('<h3>comune: '.$comune.'</h3><br/>');
echo('<h3>cap: '.$cap.'</h3><br/>');
?>
```

Fare attenzione al metodo *get* e il conseguente passaggio dei dati tra le pagine, tra l'altro la prima pagina è lato *client* e si interagisce direttamente sul proprio browser mentre la pagina invocata dalla *action* è lato server e si vedrà il risultato di elaborazione così come verrà interpretato dal Php lato *server*.

Altri esempi sono comunque presenti online in parecchi siti[12] che spiega il linguaggio anche con l'uso di tutorial. Vale la pena provare la *action* sia con metodo *get* sia con metodo *post* e notare le differenze più evidenti.

4.4 La prima query

Popolando la tabella del database direttamente da PhpMyAdmin abbiamo la possibilità di avere una tabella non vuota da interrogare da una pagina Php scritta come segue.

[11] Il protocollo HTTP (Hyper Text Transport Protocol) è argomento trattato con i protocolli di rete nel secondo quadrimestre della classe quinta.

[12] Sito in italiano utile e con esempi è www.html.it, altrimenti c'è il sito istituzionale www.w3schools.com con tutorial in inglese.

```php
<?php
   include("config.inc.php");
   /* Connessione e selezione del database */
   $connessione = mysql_connect($db_host, $db_user, $db_password);

   if ($connessione == FALSE)
   die ("Errore nella connessione.
         Verificare i parametri nel file config.inc.php");

   mysql_select_db("Sql1", $connessione)
   or die("Selezione del database non riuscita");

   $query = "SELECT * FROM studente ORDER BY cognome, nome";

   $risultato = mysql_query($query)
   or die("Query fallita: " . mysql_error() );

   /* Stampa dei risultati in HTML */
   echo('<table border="1" cellpadding="5">\n');
   echo('<tr><td>Nome</td><td>Cognome</td>
         <td>indirizzo</td><td>Comune</td><td>Cap</td></tr>');
   while ($riga = mysql_fetch_assoc($risultato)) {
      echo('<tr>');
      echo('<td>'.$riga["nome"].'</td>');
      echo('<td>'.$riga["cognome"].'</td>');
      echo('<td>'.$riga["indirizzo"].'</td>');
      echo('<td>'.$riga["comune"].'</td>');
      echo('<td>'.$riga["cap"].'</td>');
      echo('</tr>');
   }
   echo "</table>\n";

   /* Libera risorse del risultato */
   mysql_free_result($risultato);
   /* Chiusura della connessione */
   mysql_close($connessione);
?>
```

Se non si conoscono le istruzioni Php da utilizzare per la connessione al database le si può cercare in Internet[13]. L'interrogazione al database MySQL deve avvenire tramite opportuni passaggi quali:

- connettere il database, tramite istruzione *mysql_connect* con opportune credenziali di accesso (username e password) per garantire la sicurezza;
- selezionare lo schema desiderato, quello dove si trova la tabella da interrogare, tramite l'istruzione *mysql_select_db*;
- definire una stringa per la query ed eseguire la query stessa tramite l'istruzione *mysql_query*;
- eseguire un ciclo che scandisca il *recordset* del risultato fino a che sono presenti record tramite l'istruzione *mysql_fetch_assoc* che restituisce un record (riga della tabella) alla volta;
- leggere i campi del record che si vogliono stampare;
- rilasciare la risorsa utilizzata per il *recordset* con l'istruzione *mysql_free_result* e chiudere la connessione con l'istruzione *mysql_close*.

In questo caso emergono gli aspetti di programmazione e la necessità di conoscere l'opportuna libreria del linguaggio e i metodi utili alla scopo.

E' preferibile fare uso di istruzioni che catturino le segnalazioni di errore, come l'istruzione *mysql_error*.

4.5 Il primo comando

Dopo la query è conveniente proporre un esempio di esecuzione di una istruzione Sql, in questo caso si utilizzerà la INSERT in modo da popolare la tabella direttamente utilizzando l'applicazione. Si può quindi alterare il file *form_action.php* precedentemente proposto, ora nel file non si leggeranno solo i dati provenienti dalla form ma li si userà per eseguire un'istruzione *Insert* sulla tabella, in tal modo la tabella verrà popolata con una riga aggiuntiva.

[13]Informazioni sulla libreria con esempi si trovano al sito http://www.php.net/manual/en/ref.mysql.php.

```
<?php
$nome = $_GET['fnome'];
$cognome = $_GET['fcognome'];
$indirizzo = $_GET['findirizzo'];
$comune = $_GET['fcomune'];
$cap = $_GET['fcap'];

include("config.inc.php");
/* Connessione e selezione del database */
$connessione = mysql_connect($db_host, $db_user, $db_password);

if ($connessione == FALSE)
die ("Errore nella connessione.
      Verificare i parametri nel file config.inc.php");

mysql_select_db("Sql1", $connessione)
   or die("Selezione del database non riuscita");

$query = "INSERT INTO studente (nome,cognome,indirizzo,comune,cap)
         VALUES ('".$nome."','".$cognome."','".$indirizzo.
           "','".$comune."','".$cap."')";

mysql_query($query)
or die("Query fallita: " . mysql_error() );

/* Stampa id chiave attribuito al record come
    verifica avvenuto inserimento */
printf("id attribuito %d\n", mysql_insert_id());

/* Chiusura della connessione */
mysql_close($connessione);
?>
```

Una difficoltà che potrà emergere sarà la gestione della costruzione della stringa contenente il comando da eseguire, stringa prodotta con l'istruzione di concatenazione di Php (istruzione .), infatti pur essendo le variabili interpretate come stringhe in Php nella costruzione del comando in Sql vanno apposte le virgolette singole (apostrofo ') in modo che Sql interpreti il dato

come stringa.

Riferimenti bibliografici

[1] Massimo Canducci, *PHP 5*, SEI, Torino, 2007.

[2] Paolo Camagni, Riccardo Nikolassy, *PHP Dall'HTML allo sviluppo di siti web dinamici*, Hoepli, Milano, 2005.

[3] Elizabeth Naramore, *Sviluppo di siti web con PHP5, Apache e MySQL*, Hoepli, Milano, 2005.

[4] Mark Wandschneider, *Sviluppare applicazioni web con PHP e MySQL*, Apogeo, Milano, 2006.

[5] Jay Greenspan, Brad Bulger, *Sviluppare applicazioni per database con MySQL/PHP*, Apogeo, Milano, 2004.

[6] Williams Hugh, David Lane, *Applicazioni web database con PHP e MySQL*, Tecniche Nuove, Milano, 2005.