

Basi di dati

Applicazioni web in PHP per l'interrogazione di basi di dati

Laboratorio #5

Politecnico di Torino
22 maggio 2019

Introduzione

Questa esercitazione ha come obiettivo quello di realizzare una semplice applicazione web basata su PHP per interrogare una base di dati.

1 Passi preliminari per lo svolgimento dell'esercitazione

Questa esercitazione usa il server WEB Apache e il database MySQL disponibile nel prodotto XAMPP. Al fine di poter svolgere l'esercitazione è necessario avviare entrambi i servizi.

1.1 Avvio del server MySQL sulla macchina locale e avvio di Apache

L'esecuzione degli script SQL contenenti i comandi SQL per la creazione e il popolamento della base di dati utilizzata in questa esercitazione avviene tramite l'interfaccia web di MySQL. Prima di aprire l'interfaccia web di MySQL è necessario:

- Avviare il server locale Apache
- Avviare il server locale MySQL

In particolare, eseguire i seguenti passi:

1. Avviare il programma XAMPP Control Panel
2. Avviare Apache premendo il tasto Start nella riga relativa ad Apache
3. Avviare MySQL premendo il tasto Start nella riga MySQL
4. Aprire l'interfaccia web di MySQL (phpMyAdmin) premendo il tasto Admin nella riga di MySQL (il browser si aprirà automaticamente sull'URL associata alla pagina di amministrazione e interrogazione di MySQL)
5. Per eseguire uno script SQL dall'interfaccia web di MySQL:
 - Selezionare il pannello Importa
 - Selezionare il file contenente lo script che si intende eseguire e poi cliccare su Esegui

1.2 Pubblicazione/caricamento di una pagina PHP in XAMPP

Il server web Apache presente in XAMPP, quando è attivo, usa come cartella principale la cartella *htdocs*. Questo significa che per richiedere il file *index.php*, si deve navigare all'indirizzo <http://localhost/index.php> e il file deve risiedere nel proprio filesystem al percorso `[XAMPP]/htdocs/index.php`. La cartella *htdocs* si trova nella cartella principale delle risorse di XAMPP. In Windows, di default, è salvata al percorso `C:\XAMPP\htdocs`.

2 Creazione e popolamento della base di dati

La base di dati utilizzata per questa esercitazione è la stessa che avete creato nell'esercitazione precedente. La base di dati si chiama PALESTRA e riguarda le attività di una palestra. Essa è caratterizzata dal seguente schema logico (le chiavi primarie sono sottolineate):

ISTRUTTORE (CodFisc, Nome, Cognome, DataNascita, Email, Telefono*)
CORSI (CodC, Nome, Tipo, Livello)
PROGRAMMA (CodFisc, Giorno, OraInizio, Durata, CodC, Sala)

Creare la base di dati PALESTRA e popolarla usando gli script *creaDB.sql* e *popolaDB.sql* presenti sul sito del corso. Dopo l'esecuzione degli script, le tabelle del database conterranno i dati riportati nelle Tabelle 1, 2 e 3.



Info: Per rilanciare più volte lo script di creazione/popoloamento ricordarsi di cancellare eventuali istanze del database creato in precedenza dal pannello Database oppure includere all'inizio dello script i comandi per la cancellazione delle tabelle preesistenti.

CodFisc	Nome	Cognome	DataNascita	Email	Telefono
SMTPLA80N31B791Z	Paul	Smith	31/12/1980	p.smith@email.it	NULL
KHNJHN81E30C455Y	John	Johnson	30/5/1981	j.johnson@email.it	+2300110303444
AAAGGG83E30C445A	Peter	Johnson	30/5/1981	p.johnson@email.it	+2300110303444

Tabella 1: Tabella ISTRUTTORE

CodC	Nome	Tipo	Livello
CT100	Spinning principianti	Spinning	1
CT101	Ginnastica e musica	Attività musicale	2
CT104	Spinning professionisti	Spinning	4

Tabella 2: Tabella CORSI

CodFisc	Giorno	OrarioInizio	Durata	CodC	Sala
SMTPLA80N31B791Z	Lunedì	10:00	45	CT100	S1
SMTPLA80N31B791Z	Martedì	11:00	45	CT100	S1
SMTPLA80N31B791Z	Martedì	15:00	45	CT100	S2
KHNJHN81E30C455Y	Lunedì	10:00	30	CT101	S2
KHNJHN81E30C455Y	Lunedì	11:30	30	CT104	S2
KHNJHN81E30C455Y	Mercoledì	9:00	60	CT104	S1

Tabella 3: Tabella PROGRAMMA

3 Esercizi

Assicurarsi di aver seguito gli step preliminari: la base dati deve essere presente e popolata correttamente, mentre i file PHP su cui si lavora devono trovarsi nella cartella htdocs.

Creare un'applicazione web in PHP che consenta sia l'inserimento di un nuovo corso sia l'inserimento di una nuova programmazione per uno tra quelli esistenti. I passi per lo sviluppo sono elencati di seguito.



Info: Il metodo PHP `mysqli_connet()` richiede un username e una password come argomenti. Per semplicità, ai fini dell'esercitazione, si possono utilizzare quelli di default:

- **username:** 'root'
- **password:** " (stringa vuota)

Esercizio 1 (Inserimento di un nuovo corso)

Creare un form che richiede tutti i dati necessari all'inserimento di un nuovo corso nella base dati. La pagina di risultato deve valutare il corretto inserimento dei dati da parte dell'utente verificando:

- se sono stati inseriti tutti i campi;
- se il corso inserito non è già presente;
- se i tipi di dato non sono coerenti (ad esempio il livello alfanumerico anziché numerico).

Questi controlli sono da effettuare tramite l'utilizzo di funzioni apposite.

Se l'inserimento termina correttamente, si deve visualizzare un messaggio di corretto inserimento, altrimenti si deve notificare l'errore.

Esercizio 2 (Aggiunta delle classi CSS)

Per migliorare l'esperienza dell'utente, l'esito dell'inserimento del corso deve essere stilizzato appropriatamente. Utilizzare le classi CSS per creare un banner colorato in verde/rosso nella pagina restituita dal server. Nelle Figure 1 e 3 è presente un esempio di applicazione.



Figura 1: Un esempio di inserimento riuscito.



Figura 2: Un esempio di inserimento fallito.

Esercizio 3 (Inserimento di una nuova programmazione)

Creare un form che permetta di inserire la programmazione per uno dei corsi esistenti. Il form deve permettere di scegliere, tramite due menu a tendina, il codice fiscale dell'istruttore incaricato e il codice del corso per cui si sta inserendo la programmazione. I menu devono proporre gli istruttori e i corsi già presenti nella base dati.

La pagina di risultato deve valutare il corretto inserimento dei dati da parte dell'utente verificando:

- se sono stati inseriti tutti i campi;
- se si verificano inconsistenze tra i valori indicati come orario d'inizio e di fine attività (ad esempio, ora d'inizio 18:30 e ora di fine 15:00^a),
- se i tipi di dato non sono coerenti (ad esempio il giorno indicato non è uno dei giorni della settimana).
- se si verificano problemi di compatibilità con le programmazioni già presenti. In particolare, verificare se, nel giorno e nell'ora indicate, la sala selezionata non è occupata e se l'istruttore selezionato non è occupato in un altro corso.

Questi controlli sono da effettuare tramite l'utilizzo di funzioni apposite.

Se l'inserimento non va a buon fine, la pagina deve notificare l'errore.

^aIn PHP, le date o le ore possono essere verificate (controllo del tipo) e confrontate (controllo di inconsistenze) usando la funzione `strtotime($stringa)`, che permette di convertire un orario fornito in formato stringa in uno in formato time e quindi confrontabile. Se la stringa inserita NON è in un formato corretto, la funzione ritorna FALSE, altrimenti la stringa convertita in formato time.

Inserimento nuova programmazione

Giorno	<input type="text" value="Lunedì"/>
Ora inizio:	<input type="text" value="18:00"/>
Ora fine:	<input type="text" value="19:00"/>
Sala:	<input type="text" value="3"/>
Codice istruttore:	<input type="text" value="MTTSRG41L23G273B"/>
Codice corso:	<input type="text" value="CT105"/>

Figura 3: Un esempio del form richiesto per l'inserimento di una nuova programmazione.

4 Gestione dei caratteri accentati e della codifica

La scelta della codifica dei dati testuali specifica con quale formato (ovvero, con quali e quanti byte) vengono salvati i caratteri nella base di dati. Tale scelta è fondamentale per gestire casi di caratteri non presenti nell'alfabeto inglese, come gli accenti o altri segni diacritici.

In questa esercitazione verrà utilizzata la codifica **utfmb4**, che rispetta lo standard 4-Byte UTF-8 Unicode Encoding. Il file fornito con l'esercitazione crea un base di dati che utilizza questa codifica:

```
creaDB.sql
```

```
...  
CREATE DATABASE IF NOT EXISTS PALESTRA  
CHARACTER SET utf8mb4;  
...
```

A questo punto, per poter interrogare correttamente la base dati è necessario specificare, anche nel codice PHP, la codifica che si intende utilizzare per comunicare con il server MySQL. Nella versione procedurale delle API, si usa il metodo `mysqli_set_charset()`. Il metodo può essere invocato:

```
process_request.php
```

```
...  
mysqli_set_charset($conn, 'utf8mb4');  
...
```

Infine, poichè sia il database che lo script PHP lavorano in codifica UTF-8, è **consigliabile che anche il testo sorgente HTML sia nello stesso formato**. Può succedere, infatti, che definendo dei valori staticamente che contengono, ad esempio, caratteri accentati, questi vengano codificati erroneamente. È quindi consigliabile utilizzare un editor di testo moderno (Visual Studio Code, Atom, Sublime Text ed altri) in cui è possibile specificare facilmente la codifica desiderata.